10-19-2010

A framework based on Gaussian mixture models and Kalman filters for the segmentation and tracking of anomalous events in shipboard video

Ben Wenger

A FRAMEWORK BASED ON GAUSSIAN MIXTURE MODELS AND KALMAN FILTERS FOR THE SEGMENTATION AND TRACKING OF ANOMALOUS EVENTS IN SHIPBOARD VIDEO

by
Ben Wenger

A Thesis

Submitted in partial fulfillment of the requirements of the
Master of Science in Engineering Degree
of
The Graduate School
at
Rowan University
September 24, 2010

Thesis Chair: Shreekanth Mandayam, Ph.D.

ABSTRACT

Ben Wenger
A FRAMEWORK BASED ON GAUSSIAN MIXTURE MODELS AND KALMAN FILTERS FOR THE SEGMENTATION AND TRACKING OF ANOMALOUS EVENTS IN SHIPBOARD VIDEO
2009/10
Shreekanth Mandayam, Ph.D.
Electrical and Computer Engineering

Anomalous indications in monitoring equipment onboard U.S. Navy vessels must be handled in a timely manner to prevent catastrophic system failure. The development of sensor data analysis techniques to assist a ship's crew in monitoring machinery and summon required ship-to-shore assistance is of considerable benefit to the Navy. In addition, the Navy has a large interest in the development of distance support technology in its ongoing efforts to reduce manning on ships. In this thesis, algorithms have been developed for the detection of anomalous events that can be identified from the analysis of monochromatic stationary ship surveillance video streams. The specific anomalies that we have focused on are the presence and growth of smoke and fire events inside the frames of the video stream.

The algorithm consists of the following steps. First, a foreground segmentation algorithm based on adaptive Gaussian mixture models is employed to detect the presence of motion in a scene. The algorithm is adapted to emphasize gray-level characteristics related to smoke and fire events in the frame. Next, shape discriminant features in the foreground are enhanced using morphological operations. Following this step, the

anomalous indication is tracked between frames using Kalman filtering. Finally, gray level shape and motion features corresponding to the anomaly are subjected to principal component analysis and classified using a multilayer perceptron neural network.

The algorithm is exercised on 68 video streams that include the presence of anomalous events (such as fire and smoke) and benign/nuisance events (such as humans walking the field of view). Initial results show that the algorithm is successful in detecting anomalies in video streams, and is suitable for application in shipboard environments.

ACKNOWLEDGMENTS

The author wishes to express sincere appreciation to his graduate advisor, Dr. Shreekanth Mandayam for his expertise and guidance in managing this thesis work, NAVSEA, Philadelphia for providing the grant funding for the work that this thesis is based upon, Patrick Violante for his assistance at NAVSEA in determining project objectives and acquiring training data, Dr. Robi Polikar for his excellent teachings in the relevant area of statistical pattern recognition and serving on this thesis committee, Dr. Ganesh Baliga for also generously serving on this thesis committee, as well as graduate students George Lecakes and Michael Russell for their advice and support at the Rowan Virtual Reality Lab at the South Jersey Technology Park.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

CHAPTER 1: INTRODUCTION

Since the invention of the CCD and digital imaging, digital image processing has been used to analyze and enhance quantized images. As technology progressed, the natural extension of machine vision research was the analysis of sequences of images. Video recording systems have proved to be an excellent method of providing remote surveillance, simply by their basic nature. Due to decreasing costs of such systems, they have been used in nearly all places and applications where they may be beneficial. Computer aided video analysis is of utility to nearly all types of industries and it is particularly useful to those involving any kind of human visual analysis.

The US Navy has recently strived to develop technologies and strategies to reduce manning in DDG 51 class warships. This is mindful of future classes of naval ships that will benefit directly from methods developed for the current fleets. Cost reduction is a primary reason for this movement. Perhaps a more important reason is that more capable crews, often operating newer-technology systems, can make more effective decisions in a more-timely manner. Also, smaller crews can free up resources to help recapitalize the naval fleet [1]. The cost reduction initiative comes directly from investigation of the Navy budget since 1985. The total operating budget has declined by 40% (as well as ship count by 45%), but the operations and support costs (directly personnel related costs) have not

decreased. These costs have continued to grow in spite of the smaller fleet and reduced budget [2].

A US Navy study [1] gives specific mention to "the use of emerging technologies, such as the multi-modal watch station consoles" allowing "a single watch stander to take on several roles from the one console. In testing, the improved situational awareness allowed the multi-modal teams to fare better than the control team." One particular aspect of these consoles is video surveillance. Surveillance video analysis by a computer system allows a station console watcher to be alerted of a detected unusual event and immediately gain access to the video feed in question. A capable system would be even more helpful when combined with a distance support initiative. Using allocated bandwidth for ship to shore communications, a video analysis system could relay notice of detected events on ships around the world to remote operators. In this way, very few people are required to determine the severity of video analysis results. Video feeds would only need to be viewed in the event of a detected problem rather than at all times as in a traditional surveillance system.

1.1 Applications of Video Analysis

Video analysis is a broad area of research due to the ubiquity of digital video and an ever-expanding number of ideas for potential ways that it may mimic human vision. The field of machine vision can be split into many different areas of concentration, but video analysis carries some particularly popular applications. Many of these applications are specific types of event detection, combining image characteristics with time as a very important

additional variable compared to basic imagery. The applications are not unique to other technologies in that they primarily seek to automate processes that are prone to human error, are costly to provide human supervision, or may give substantial benefit to existing human operated computer systems.

1.1.1 Human Motion Analysis

An obvious application of video analysis is human motion analysis. There are countless examples of how a system capable of recognizing and categorizing human motion may be useful. Surveillance video of human positions over time may determine unusual events indicative of threats to security. Tracking positions of human limbs can be used as a biometric in combination with facial recognition for automatic identification of individuals from a distance. Gesture recognition is also an excellent example of human motion analysis, as it combines recognition of specific parts of the body with their positions over time.

1.1.2 Traffic Analysis

The analysis of city and highway vehicle traffic is perhaps the most popular and accurate use of machine vision for video analysis. Long stretches of highway cannot be practically monitored by humans alone. Video has the advantage of a large field of view (FOV), allowing strategically placed cameras to simultaneously cover an entire highway. Vehicle movement is not very complex; therefore it is not difficult to determine when vehicles move away from their normal course on a road. Stopped or abnormal vehicle movement

is one of the primary utilities of a traffic analysis system. Law enforcement can be immediately alerted of a detected vehicle break-down or accident.

Speed of vehicles can also be detected, and in combination with the recognition of vehicle license plates using optical character recognition (OCR), automatically issuing speeding tickets to drivers is a real and tested capability. The relative flow of traffic can also be determined from the speed and quantity of moving vehicles in a section of highway. This is a good example of how a video sensing system can provide a quantitative output as a measurement, similar to any traditional sensor which may sense temperature, pressure, or light for example.

1.1.3 Military Weapon Systems

The use of video for assistance in military applications is one of the oldest known applications of video sensing technology. Using video as the eyes for a computerized weapon system to track onto targets is a major capability. Video analysis can also be used for many types of military surveillance purposes, including site surveillance. Similar to traffic analysis as previously discussed, a video camera pointed at an enemy base for example can provide a way of automatically determining the areas of most activity. Over time, the activities could even be categorized as desired to be normal, indicative of some sort of preparation for attack, or some other notable condition. Other types of military surveillance systems involve heavy video analysis including Unmanned Aerial Vehicle (UAV) systems. UAV surveillance systems are particularly complex due to their non-stationary nature. Location and direction must be combined with analysis of moving imagery in order to properly make sense out of all terrain and visuals that are seen.

1.1.4 Video Categorization Systems

A video analysis system can be trained to look for certain types of tracked motion and behaviors of scene objects in videos of known activities. Given enough videos of typical motion and behaviors, it can be determined what general activity is occurring in a newly presented sequence. This can be used to automatically categorize large databases of videos into specific subsets. Internet databases of videos would be most capable of utilizing this type of system, as they are very large and it is simply not practical to have human operators categorize every video in a database.

1.1.5 Unusual/Anomalous Event Detection

A recurring theme in video analysis is the use of anomalous/unusual event detection. As motion and scene characteristics are identified, they can be categorized in a way that allows future conditions to be compared to what occurs most often in a scene. Probabilistic maps of scenes are created and iteratively updated that show where activity is likely to occur. Creating a threshold of probability for a certain condition to occur will allow for the detection of an unusual event, no matter what it may be specifically. The important factor is that the scenario has not occurred previously, or at least not very often.

A perfect example of video anomaly detection is shown in the application of detecting smoke and fire. Instead of using traditional smoke or fire detection equipment, image processing can be employed to detect characteristics of smoke and fire. There are a few advantages of this method, the most important being the increased range of detection using a video sensing system. A standard smoke alarm sensor in a large atrium

or lobby for example may take several minutes to detect a problem because the smoke must rise all the way to the sensor. If a video-based smoke/fire sensor were employed, the time for detection could be decreased to just seconds. Time is incredibly valuable in the case of a fire, thus development of methods to quickly determine the presence of fire is of great importance.

1.2 Motivation

The major incentive for this thesis has spurred from the need for technologies to decrease manning in shipboard environments. Automated surveillance reduces the amount of time needed to be dedicated for the human element of video surveillance systems. Remotely operating the automated surveillance systems from a base in-land will reduce the amount of people needed even further. Moving jobs that previously could only be done on a shipboard system to a centralized location frees up workers within the entire fleet. Costs are reduced immediately by lowering demand for personnel on ships who can perform the same work at a land-based operation.

There is a reason for applying a full object tracking system to solve this problem rather than looking at independent frame image characteristics alone. It is believed that specific spatiotemporal features unique to smoke and fire requiring tracking will be advantageous to their swift characterization. Some general features that smoke and fire tend to exhibit are steady growth in size and slow movement. In order to determine these features, it must be known which objects match each other between frames of video. Once this is determined, then various temporal characteristics may be extracted.

1.3 Objectives and Scope of the Thesis

This thesis focuses on the design, development and validation of algorithms for the detection and tracking of anomalous events that can be identified from the analysis of monochromatic stationary ship surveillance video streams. The specific anomalies that we have focused on are the presence and growth of smoke and fire events inside the frames of the video stream. The objectives of this thesis are to:

1. Compile a survey of existing techniques for analyzing shipboard video stream data;
2. Design and develop a video foreground segmentation algorithm for determining regions of interest in video streams;
3. Design and develop an object tracking system capable of persistently tracking objects between frames;
4. Identify distinct and robust features from the tracked objects for detection and classification of anomalous indications;
5. Exercise the algorithm on a database consisting of canonical and experimental videos streams embedded with known anomalies (smoke and fire) as well as benign content;

The data that has been analyzed in this thesis consists of experimental video streams compiled from online sources, including fire protection engineering tests, video surveillance data for human motion analysis, stock video footage, as well as video streams made available through other academic research projects. These video streams contain anomalous indications representative of smoke and fire events, as well as regular human motion representative of nuisance events. Validation of the algorithms developed in this thesis will be based on their performance of classification between anomalous and nuisance events within this developed experimental video database.

1.4 Organization of the Thesis

The work performed within this thesis will determine how well a general foreground segmented object tracking system will perform when applied to the detection of smoke and fire in video streams. The system is geared toward but not specific to placement within shipboard environments. Training and testing is limited to available video sequences, including publicly available fire training videos, laboratory videos, surveillance video analysis database videos, select examples of anomalous sequences from internet sources, computer generated anomalous sequences, as well as nuisance surveillance data from the NAVSEA test facility in Philadelphia, PA.

This thesis is organized as follows. Chapter 1 introduces the concept of video analysis and its use in various applications as well as the motivation for this thesis. Chapter 2 gives an overview of previous work in the area of video analysis, some background for the fire and smoke detection technologies this work intends to substitute/improve upon, information about shipboard systems, and background information for essential automated video surveillance system components. Chapter 3 provides a description of the approach used to perform the video analysis tasks including foreground segmentation, morphological processing, Kalman filtering, feature extraction, Principal Component Analysis (PCA), and neural network classification. Chapter 4 presents a discussion of results of performance of foreground segmentation, object tracking, and classifier performance on the training data set. Chapter 5 contains a conclusions of the work completed in this thesis with recommendations for future work in this area.

CHAPTER 2: BACKGROUND

In order to gain an understanding of the methods and principles behind the work presented in this thesis, it is necessary to cover a background in video analysis as well as fire detection. A summary of previous work in video analysis as well as fire detection is provided in this chapter, outlined in Table 1 and discussed in the following sections. A discussion of material pertaining to smoke and fire detection methods as well as their role in shipboard systems is also provided, followed by an in-depth overview of video analysis system components.

2.1 Previous Works

Table 1. Previous works in video smoke/fire detection

Author(s)/ Publication	Title of Work	Summary
Schultze, Kempka, Willms, *Fire Safety Journal* [3], 2006	Audio-Video Fire-Detection of Open Fires	Detected flickering frequency of fire based on analysis of high speed black and white video as well as audio recordings.
Neal, Land, Avent, Churchill, *American Research Corporation of Virginia* Report [4], 1991	Application of Artificial Neural Networks to Machine Vision Flame Detection	Detected flames in video using simple HSI (Hue Saturation Intensity) filtering and thresholding for features to be used for training of neural network classifier. Results were highly accurate for the training and test data set.
Töreyin, Dedeoğlu, Güdükbay, Çetin, *Pattern Recognition Letters* [5], 2006	Computer Vision Based Method for Real-Time Fire and Flame Detection	Detection of fire is based on temporal and spatial wavelet analysis of moving pixel regions with thresholded pixel colors. High frequency activity in the segmented regions were classified based on thresholds in variation of wavelet transform coefficients per pixel.

Ko, Cheong, Nam, *Fire Safety Journal* [6], 2009	Fire Detection Based on Vision Sensor and Support Vector Machines	A temporal differencing method of motion detection as well as fire-colored pixel detection was used to extract regions for which to calculate wavelet transform coefficients. The coefficients represented a fire model used for Support Vector Machine (SVM) classification. Results show that it is a more robust method of fire detection than the work it had meant to improve upon.
Han, Lee, *Fire Safety Journal* [7], 2009	Flame and Smoke Detection Method for Early Real-time Detection of a Tunnel Fire	A video-based fire and smoke detection algorithm is introduced with emphasis on speed of detection and application for highway tunnels. Color and motion are used to segment regions against a background image of the scene based solely on thresholding. High performance results were posted for fire and smoke detection on a small testing set.
Ono, Ishii, Kawamura, Miura, Momma, Fujisawa, Hozumi, *Fire Safety Journal* [8], 2006	Application of Neural Network to Analyses of CCD Colour TV-Camera Image for the Detection of Car Fires in Expressway Tunnels	A reference background image was used for RGB color thresholding to determine fire regions. A neural network was used to classify fire from RGB histogram quartile values as well as detected flame area.
Ho, *Measurement Science and Technology* [9], 2009	Machine Vision-Based Real-Time Early Flame and Smoke Detection	Detects fire and smoke through spectral, spatial, and temporal characteristics. Tracking in video is implemented through a motion history technique combined with the CAMSHIFT algorithm. Histogram analysis using fuzzy logic, turbulence measurement, temporal flickering analysis
Kim, Wang, *Proceedings of the 2009 World Congress on Computer Science and Information Engineering* [10], 2009	Smoke Detection in Video	Smoke is detected based on analysis of motion segmented regions determined against a background reference image. The color and shape features of the motion regions are kept for a window of time and smoke is classified from the changes between the features at each frame.
Calderara, Piccinini, Cucchiara, *Computer Vision Systems* [11], 2008	Smoke Detection in Video Surveillance: A MoG Model in the Wavelet Domain	Algorithm was developed to detect smoke based on analyzing motion segmented regions using analysis of wavelet transform coefficients. A mixture of Gaussians approach is used to classify smoke from non-smoke coefficients and color and textural information.

Gubbi, Marusic, Palaniswami, *Fire Safety Journal* [12], 2009	Smoke Detection in Video Using Wavelets and Support Vector Machines	A block based approach to analyzing video using the wavelet transform was used with an SVM classifier determining smoke from non-smoke in each block. The performance of DCT coefficients were also analyzed similarly. Results from analysis on the green image channel for forestry background video was accuracy near 90% with high specificity and sensitivity.
Gómez-Rodríguez, Arrue, Ollero, *Proceedings of the Eighth SPIE Automatic Target Recognition Conference* [13], 2003	Smoke Monitoring and Measurement Using Image Processing: Application to Forest Fires	Wavelet based optical flow calculation used for segmentation of smoke regions in forest surveillance videos. Proposes the use of optical flow for monitoring smoke characteristics in order to classify based on detected movement and growth.
Chunyu, Jun, Jinjun, Yongming, *Fire Technology* [14], 2009	Video Fire Smoke Detection Using Motion and Color Features	Algorithm developed which combines temporal differencing motion detection, color segmentation, and optical flow motion estimation. Optical flow motion outputs are used for training of a backpropagation neural network architecture for detection of smoke in videos.
Liu, Ahuja, *Proceedings of the 17th International Conference on Pattern Recognition (ICPR) 2004* [15]	Vision Based Fire Detection	Proposed a method for detecting flame regions by modeling the stochastic nature of the contour of those regions. The variability is modeled using Fourier descriptor coefficients with an autoregressive model in order to train an SVM for classification.

The work of Schultze et al. [3] describes a method of detecting open fires based on an analysis of the flickering frequency of fire. This flickering frequency is determined both by audio and visual analysis, using video and audio recordings of test fires. A high speed black and white video camera and sensitive microphone were used for the recordings. The goal of video analysis was to determine the flickering frequency and flow movement of ethanol test fires. It was determined that flickering frequencies of fires can be seen from both video and audio analysis, always below 10Hz. Estimation of the motion was also calculated using a block-based motion estimation method to determine the flow of

movement in video, simplified to only vertical motion using analysis from a standard video recording device rather than a high speed device. Results show that the frequency and motion analysis of flames based on their flickering is a possible characteristic for classification, but effects of nuisances and noise in the data has not been evaluated.

A US Air Force funded study [4] has demonstrated a very simplistic yet effective method of detecting flames in color videos. The reasoning for the study was to determine the feasibility of an image processing based approach as a faster and more accurate method of fire detection than existing technologies. The method that was used involved thresholding of low pass filtered video data by extracting hue and saturation values indicative of fire. Also used as a feature for fire severity of thresholded regions was the growth of the region over time. The hue and saturation patterns from detected regions were extracted from training videos and then used for training of a neural network algorithm to classify true flame videos from false alarm (nuisance) videos. The result of this system was 100% accuracy on the test video data set (17 fire, 6 nuisance), with accurate detection of an approximate 4ft x 4ft area at a range of 150 feet. Test and training videos were limited to JP4 jet fuel burning videos in airplane hangars. This study proved that a neural network based image processing system could be successfully employed to detect fire based on region segmented video analysis. The use of growth characteristics was also proven to aid in the detection of fire regions as well. The system is completely reliant on color space processing, however a similar method of analysis could be done using a different segmentation criterion.

Töreyin et al. [5] developed a flame detection system that utilized a combination of motion segmentation and pixel color regions. These regions become input to generate wavelet coefficients in the time and spatial domain to be used for final classification. The motion detection (foreground segmentation) method was based on a weighted average of previous frames as a background model and thresholded based on the difference from the background model as discussed in [16]. The moving pixel regions were then further filtered based on their color values being within a certain threshold indicative of the color of fire and flame regions. The remaining pixels were subjected to the wavelet transformation for analysis of high spatial frequency content and temporal frequency related to flickering flames. Thresholding of the analyzed wavelet coefficients yielded the final classification of fire for the detected region. The results of this system are very good for the videos used for testing, with a stated false positive rate of 0.01%. The work claims to have a working speed of real-time for 320px x 240px videos at 10 frames per second (FPS) or higher.

Ko et al. [6] created a fire detection system based on modifications of the work by Töreyin et al. [5]. The improvements of the system were on two different levels. First, instead of using a simple threshold, pixels representing colors determined to be representative of flames were extracted based on probability estimates from training examples in the motion detected regions. These regions were then treated to the wavelet transformations as in [5], but with the coefficients stored as features for training of the SVM classifier rather than used for thresholding. Results of this study show that compared to the previous method, the proposed system is more robust to noise and differences

between frames due to the additional pattern recognition techniques. However, the system still needs improvement in detection accuracy and computation time for a real-time fire detection system.

Han and Lee [7] introduced a video-based fire and smoke detection algorithm with emphasis on speed of detection and application for highway tunnels. The algorithm uses differences in intensity and color to segment fire regions against a background image of the scene. Motion detected over a background image was used similarly to segment smoke regions, with invariant image moments calculated over each one. Invariant image moments are discussed in detail in section 2.7.3.1 of this thesis. Both techniques classified objects based solely on thresholding. High performance results were posted for fire and smoke detection on a small testing data set.

Ono et al. [8] developed an algorithm for detecting fire in expressway tunnels using test data created to mimic a tunnel environment. A reference background image was used for RGB color thresholding to determine fire regions. Red and green color relationships were found to be of considerable importance. A neural network was used to classify fire from RGB histogram quartile values as well as detected flame area, with a thresholded neural network output determining true fire detection. Some background on neural network classifiers is given in section 2.8.2 of this thesis. Experimental results showed that the system worked well with near 100% performance for various detection distances for a tray fire. Although the system used color data, it provides some evidence of the applicability of neural networks to video image detection technology, in particular for fire classification.

The work by Ho [9] is perhaps the most similar to the algorithm presented in this thesis. The author's algorithm was able to detect fire and smoke through spectral, spatial, and temporal characteristics. A motion history technique was used to determine regions most probable to contain smoke or fire, which were then integrated with a level crossing rate (LCR) threshold. Tracking in video was implemented using the CAMSHIFT algorithm, which works by analyzing probabilities of the image histogram over time. Histogram analysis using fuzzy logic, turbulence measurement, and temporal flickering analysis were combined for a final decision method. The performance of the algorithm was comparable to previous works for the tested videos, however it was not considered to be superior to alternative methods. The author seeks to continue the work by using a neural network for classification in lieu of the LCR threshold detection method.

Kim and Wang [10] demonstrated a simplified approach to smoke detection with application to detection in forestry surveillance video. A background reference image is kept in order to calculate differences in the current frame. The color and shape features of the detected motion regions are kept for a window of time and smoke is classified from the changes between the features at each frame. The paper showed a need for improvement of the technique along with additional testing. This demonstrated the difficulty of the smoke detection problem, and that the simplest way of tackling the problem is not the best. There are a variety of potential problem factors that have not been addressed, including accuracy of motion detection and environmental factors of outdoor video surveillance.

Calderara et al. [11] provided work in the area of smoke detection within color videos based on the method of Töreyin et al. [5]. They developed a method to detect smoke based on analyzing motion segmented regions using analysis of discrete wavelet transform (DWT) coefficients. Instead of using a threshold [5] or SVM classifier [6] on the DWT coefficients, a mixture of Gaussians approach is used to learn the distribution of DWT energy in smoke regions. Color and textural information is differentiated in this way. A color blending function is used to determine the relationship between the motion segmented regions and a reference smoke color model. A Bayesian probability is maintained at a block level throughout the image in order to determine likelihood of smoke based on the calculated DWT energy as well as color blending function output. Performance of the algorithm on a variety of over 50 test videos shows its effectiveness. Since it relies on the time-based variation of the DWT coefficient energy, the detection rate increases from 77% after 3 seconds, to 98.5% after 6 seconds, and finally to 100% after 10 seconds with an average true positive smoke detection rate of 4%. This system and in general those that use DWT coefficients seem to provide some of the most reliable results for smoke detection when color imagery is available. A particular advantage of this system is that it is capable of being integrated into any type of background modeling scheme.

Gubbi et al. [12] proposed a video smoke detection algorithm with high similarity to [5], [6], [11]. The algorithm utilizes DWT coefficients calculated over 32x32px blocks in the horizontal, vertical, and diagonal directions over the first three levels of the transform. Six statistical features were calculated from the coefficients including arithmetic mean,

geometric mean, standard deviation, skewness, kurtosis, and entropy. These features were used as input to an SVM classifier along with the highest energy coefficients calculated using the discrete cosine transform (DCT) for comparison. The DCT as an image feature is discussed in detail in section 2.7.4 of this thesis. Results of testing the SVM on the DWT coefficients using only the green channel of RGB imagery was 88.75% accuracy in detection with sensitivity and specificity at 90% and 89% respectively. The results of DCT coefficients were 63.21% accuracy for SVM and 68.88% accuracy for a K-Nearest Neighbor classifier. Motion segmented imagery was used as well with the same procedure and it was found to improve accuracy to 91.47%. This work shows yet again the effectiveness of the DWT in detecting smoke in color imagery. It also provides prior merit to the use of DCT coefficients for smoke detection, which is one particular aspect of the work in this thesis.

Gómez-Rodríguez et al. [13] developed a method of measuring and monitoring the motion of smoke regions in forest fire videos. The proposed method utilized the wavelet transform method of optical flow calculation in order to segment motion regions in video. Optical flow is discussed in detail as a method of foreground segmentation in section 2.4.2 of this thesis. Based on the detected movement of segmented regions, the velocity and growth of the region can be calculated and determined to be a smoke region. The paper does not provide details on the foundation for the decision block of processing, which may be due to the nature of the application. Generally within forest monitoring videos there are few nuisance objects in the camera's field of view, which requires less complexity/necessity for an advanced decision method for determining smoke. The

outputs of the system are smoke plume height, velocity, volume, and rate of volume growth, which are representative of an image processing based measurement system applied successfully to smoke detection and quantification.

A method of smoke detection has been proposed [14] that utilized color and also primarily motion features. The regions of smoke are determined first from a color segmentation model that thresholds each pixel based on how closely it represents gray color in terms of RGB intensity. Combined with the color segmented region, the motion is estimated in the scene by using a simple temporal differencing scheme that determines the change in recent scene activity compared to a weighted average of previous frames. The final segmented region is representative of smoke objects and is subjected to optical flow calculations in order to extract its motion features. The extracted features are used to train a backpropagation neural network for real time classification of test video sequences. Results are shown in comparison to the method of Töreyin et al. [5], with similar but slightly less accuracy overall. The system as a whole shows applicability of neural networks and to smoke detection, but yet again heavily relies on color segmentation. The novel aspect of the work is the successful use of motion features from optical flow estimation as the main discriminatory feature for smoke region detection.

A novel method of detecting fire regions based on "recognizing shape evolution in stochastic visual phenomena" is proposed by Liu & Ahuja [15]. Their method relies on the recognition of random motions of contours of fire regions rather than learning any specific shapes to which fire regions may conform. Color and intensity of video frames are first used to segment bright regions. Fourier descriptor coefficients are then calculated for

these regions, and are input into an autoregressive model for capturing multiple levels of temporal variation. Fourier descriptors represent a spectral representation of the contour of a 2D region by splitting the x and y coordinates of the contour into real and imaginary components of a complex signal. The complex signal is treated to the Fourier transform, allowing shape information to be stored in the frequency content of the transform coefficients. The coefficients and autoregressive models from training data sequences are used for the creation of an SVM classifier with a radial basis kernel. Results from tests on a variety of sequences show the algorithm's effectiveness even in light of camera motion.

Table 2. Previous works in surveillance video analysis and event detection

Author(s)/ Publication	Title of Work	Summary
Cristani, Bicego, Murino, *IEEE Transactions on Multimedia* [17], 2007	Audio-Visual Event Recognition in Surveillance Video Sequences	Developed algorithm to detect events in video by integrating scene color information as well as recorded audio. The information is combined and input into a KNN (K Nearest Neighbors) classifier.
Brand, Kettnaker, *IEEE Transactions on Pattern Analysis and Machine Intelligence* [18], 2000	Discovery and Segmentation of Activities in Video	Through the use of Hidden Markov Models (HMMs), a system was developed that allows observed activity to be organized into meaningful states. Experiments on videos of office activities show its effectiveness in segmenting video into scenes determined by even very minute differences.
Medioni, Cohen, Brémond, Hongeng, Nevetia, *IEEE Transactions on Pattern Analysis and Machine Intelligence* [19], 2001	Event Detection and Analysis from Video Streams	Detected the movement of objects from an Unmanned Aerial Vehicle (UAV) in the presence of camera motion using optical flow. Objects are tracked and analyzed using a graph-based approach with dynamically updated templates for object trajectories. Behavior analysis is completed successfully using detected motion and is applied to Predator UAV test video sequences.

Xiang, Gong, *IEEE Transactions on Pattern Analysis and Machine Intelligence* [20], 2008	Video Behavior Profiling for Anomaly Detection	Using mixtures of dynamic Bayesian networks, anomalous activities in video are detected by subjection to a likelihood ratio test. The probability based measure is robust to noise and is adaptable so that anomalous conditions may become normal conditions over time. Test results on office corridor activity showed improved performance compared to a supervised equivalent algorithm requiring offline training.
Koller, Weber, Huang, Malik, Ogasawara, Rao, Russell, *Proceedings of the 12th International Conference on Pattern Recognition 1994* [21]	Towards Robust Automatic Traffic Scene Analysis in Real-Time	Tracking of vehicle traffic is performed using stationary camera surveillance. Vehicle motion is segmented from the background using a Kalman filter based method, with blob analysis and position tracking performed in the steps that follow. A dynamic belief network is used to process the high level reasoning of the scene. Performs well under ideal conditions, but may require additional development for real-time capability and desired robustness.
Collins, Lipton, Kanade, Fujiyoshi, Duggins, Tsin, Tolliver, Enomoto, Hasewaga, Burt, Wixson, *American Nuclear Society (ANS) Eighth International Topical Meeting on Robotics and Remote Systems* [16], 1999	A System for Video Surveillance and Monitoring	Comprehensive stationary video surveillance system for application in many applications including object classification, vehicle tracking, gait analysis, and airborne object tracking. The system is very complete in its ability to calibrate and integrate multiple camera views of a single scene and tie in geolocation capabilities in differing applications. Motion segmentation, object tracking, trajectory filtering, neural network object classification, and activity recognition have all been integrated into the algorithm.
Haritaoglu, Harwood, Davis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* [22], 2000	W^4: Real-Time Surveillance of People and Their Activities	Full, automated, surveillance system with application to human detection and activity analysis in monochromatic video sources. Capable of detecting moving people/objects, distinguishing people from other moving objects, classifying known object types, tracking multiple people simultaneously during interaction, tracking body parts, and detecting events relating to person/object interaction.
Xiang, Gong, Parkinson, *Proceedings of the British Machine Vision Conference 2002* [23]	Autonomous Visual Events Detection and Classification Without Explicit Object-Centred Segmentation and Tracking	Developed a method of detecting events in video without using the traditional method of tracking moving objects. Only the pixel-level motion is considered at each frame with a minimum connected component size. A pixel change history is maintained and used for generating features to be clustered using the expectation maximization (EM) algorithm. Results on grocery shopping sequence show potential as well as need for improvement.

Zhang, Li, Huang, Tan, *Proceedings of the 19th International Conference on Pattern Recognition 2008* [24]	Boosting Local Feature Descriptors for Automatic Objects Classification in Traffic Scene Surveillance	Used AdaBoost algorithm on shape feature descriptors for discrimination between people and vehicles in traffic surveillance videos. The HOG, SIFT, Spin Image, and RIFT descriptors are experimentally applied to determine effectiveness on hand labeled training data. SIFT descriptor was determined to have the greatest performance on training data and in combination with SPIN feature on data fusion evaluation of training data as well.
Gryn, Wildes, Tsotsos, *Computer Vision and Image Understanding* [25], 2009	Detecting Motion Patterns Via Direction Maps with Application to Surveillance	Proposed the use of an energy-based spatiotemporal direction map for detection of events in surveillance videos. Features of global motion are extracted and compared to templates of previously defined direction maps. Detection of events was 90.78% in test videos of human and traffic surveillance sequences.
Jäger, Knoll, Hamprecht, *IEEE Transactions on Image Processing* [26]	Weakly Supervised Learning of a Classifier for Unusual Event Detection	Hidden Markov Models are used in an incremental fashion to learn patterns of activity in image sequences. Principal component eigenvectors from raw image sequences are used as features for classification as well as the distances in Euclidean space from the principal components. Results on laser welding image sequences show effectiveness of the algorithm with a small estimated false positive rate of 1.8%.
Stauffer, Grimson, *IEEE Transactions on Pattern Analysis and Machine Intelligence* [27], 2000	Learning Patterns of Activity Using Real-Time Tracking	Developed a system for analyzing surveillance video of a parking lot for detection of particular activities. Motion segmentation is performed using a novel adaptive mixture of Gaussians model and applied to a predictive tracker based on linear Kalman filtering. Relative camera geometry is used to model the scene structure based on homography between cameras. Basic shape features are used to distinguish people from vehicles and their tracked positions are then clustered to determine normal/unusual site activities.
Basharat, Gritai, Shah, *Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition* [28], 2008	Learning Object Motion Patterns for Anomaly Detection and Improved Object Detection	Proposed the use of Gaussian Mixture Models (GMMs) for learning positional and trajectory information on a per-pixel basis within a surveillance scene. EM is used to compute GMM parameters in an online fashion when enough tracking information is collected. Anomalous motion activity is detected using stored trajectory distribution information. A novel background model learning rate adjustment is introduced as well. Testing results show effectiveness in detecting untrained anomalies in speed and trajectory of tracked objects as well as success in improving the background update model using size/velocity tracking features.

Cristani et al. [17] described a method for combining both audio and video features together in order to detect changes in a scene. A time adaptive Gaussian mixture model is used for per-pixel changes, and a positive histogram change analysis is used to detect the start of a change in a scene. A single microphone is used for audio analysis, with a Gaussian mixture model also used for change detection of audio features such as the subband energy amount, which represents the histogram of the spectral energy with each bin representing a certain interval of time. The audio-visual data is fused by looking at the histogram pair at each time interval for the audio based histogram and the video frame based histogram. Changes in this data (called an AVC matrix), determine the total duration of an event for video segmentation. The AVC matrix information can be used to classify an event as well, using KNN and clustering for classification. The concatenation of both audio and visual features allowed for increased accuracy in classification by KNN as well as clustering than for each feature alone. Overall performance was 89% for differentiating six different events in a segmentation of 66 video sequences from a two hour video in an office setting.

Brand & Kettnaker [18] proposed an algorithm that utilized Hidden Markov Models for the automated segmentation of activities in video sequences. They used a minimization of an entropy parameter for development of the HMMs. The features used for the learning of scene activity were based on the shapes of "blobs" or connected components calculated from a foreground segmentation method. The system has only been tested thus far on the activities of a single object moving in the scene. The algorithm is also capable of detecting anomalous scene activities, but cannot determine exactly

what type of activity is occurring. Results are considered successful for various types of data including office activity, intersection traffic, and human motion capture: however they are largely qualitative and depend on the application.

Medioni et al. [19] discussed an event detection system which was developed specifically for the application of handling UAV video streams. The first processing component includes motion detection in the presence of camera motion through the use of differential-based optical flow estimation. The detected motion allows objects to be tracked and described using motion features. Such attributes of each region/node can be used for clustering or template matching from centroid, principal direction (eigenvector), mean, variance, velocity, and length. Tracked objects also have features of height, width, speed, motion direction, and the distance to another reference object. Behavior analysis of these features is conducted over time, and in the context of various scenarios can be detected with accuracy dependent upon the quality of the source imaging data. The system has proven to be effective for vehicle tracking and analysis using data from the Predator UAV.

Xiang and Gong developed an unsupervised algorithm [20] that learns scene behavior in surveillance video by dynamically updating Bayesian networks. The abnormality of the behavior is determined by employing a likelihood ratio test. The probability that an event/behavior has occurred is updated by using extracted features of motion and shape in scene objects including centroid position, width and height of bounding box, bounding box fill ratio, and pixel change history within the moving object bounding box. An online EM algorithm is used to update the dynamic Bayesian network. A

description of EM is given in section 2.4.3.1. Results of the system were based on ground truth from hand labeling the behaviors in an office corridor sequence. The proposed algorithm was compared to an equivalent algorithm which used a supervised batch learning method. Detection rate was higher and false alarm rate was lower for the proposed algorithm in trials using the test data, and it also had a lower computational cost than the supervised algorithm.

One particular system utilizing analysis of traffic flow in a highway situation is presented by Koller et al. [21]. A variety of methods employed in a typical surveillance application are used for highway scene analysis. A Kalman filter based motion detection algorithm is used to segment the moving vehicles from the background. The vehicle positions are tracked even in the presence of occlusion due to prior knowledge of the scene and the primarily linear object trajectories. Overall scene analysis is achieved through the use of dynamic belief networks which assign probabilities to events in a hierarchical structure. This allows certain events to be detected such as lane changes and stalled vehicles. The system has been tested using real-world four lane highway surveillance data and is successful in tracking 8-10 vehicles simultaneously and detecting events using the dynamic belief network architecture.

The work of Collins et al. [16] is one of the most complete video surveillance systems that has been developed at this time. The system ties together all of the main facets of an automated surveillance system including a motion segmentation algorithm, tracking using position filtering and prediction, object classification, and activity recognition all with improved accuracy through multi-camera calibration and site

modeling using geolocation. The system has been applied for applications of vehicle tracking, gait analysis, and airborne object tracking. The method of motion segmentation is a hybrid using three-frame differencing. Tracking is based on recognizing objects between frames based on recognized shape, color, and positional features. The detected object trajectories are integrated into activity recognition for solitary and interacting objects using statistical Markov models.

The algorithms developed by Haritaoglu et al. [22] are highly capable of determining human and human/object interaction within monochromatic video. The system is proficient in detecting moving people/objects, distinguishing people from other moving objects, classifying known object types, tracking multiple people simultaneously during interaction, tracking body parts, and detecting events relating to person/object interaction. Motion segmentation is achieved by comparing video frames to a bi-modal Gaussian model of the scene background created during an offline training period. Moving regions are further processed by noise cleaning, morphological filtering, and object detection. Textural and shape information is maintained for each detected person for tracking during occlusion, and a silhouette model is used for assumptions on human shape. Even when two or more people are in a group, individuals can be tracked independently.

The method of event detection proposed by Xiang and Gong [23] is one that seeks to avoid the ill-posed problem of moving object tracking and identification. Instead it seeks to cluster features of connected components segmented by an adaptive mixture of Gaussians background modeling method. A history of the amount of change in image

regions kept over time represents a spatiotemporal model of a potential event. The EM algorithm is used to cluster the features extracted from a training sequence. Experiments on a video of grocery shopping revealed that the system is capable of discriminating unique events but fails to post good performances, especially during any complex motion or occlusion.

Zhang et al. [24] examined the performance of local feature descriptors in order to distinguish between people and vehicles in traffic surveillance videos. Motion segmentation using a mixture of Gaussians was employed and from the segmented objects, four different powerful feature descriptors were extracted. The HOG (Histogram of Oriented Gradients), SIFT (Scale Invariant Feature Transform), Spin Image, and RIFT (Rotation Invariant Feature Transform) feature descriptors were evaluated on a large database of hand-labeled people and vehicle images. The AdaBoost algorithm was used with an unspecified base classifier. AdaBoost, which is short for adaptive boosting, is a classifier ensemble training algorithm that adapts during the training process, to give increasing influence to those training data instances that are more difficult to classify. Highest performance on the training data was 98.2% for vehicle classification and 99.3% for people classification using SIFT features. A fusion of paired combinations of features was examined as well, and it was determined on test data that the SIFT/Spin Image feature combination attained the best performance.

Another method of event detection in video is presented by Gryn, Wildes, and Tsotsos [25] that is similar to [23] in that object motion is not explicitly detected tracked as in traditional surveillance video analysis methods. The algorithm works on global

motion calculation determined from an energy-based calculation using multiple spatiotemporal filters. The detected features are represented geometrically in (x, y, t) space as normal vectors of the faces of a dodecahedron. The result is a direction map similar to an optical flow estimation that can be matched to pre-computed direction map templates. Testing on videos of human and traffic surveillance resulted in 90.78% successful detection.

Jäger et al. [26] proposed an application of Hidden Markov Models used in an incremental fashion to learn patterns of activity in image sequences. Principal component eigenvectors from raw image sequences are used as features for classification as well as the distances in Euclidean space from the principal components. A regular sequence model is generated from known benign data. Following this step, an error sequence model is created based on outlier detection from the regular model in order to detect future unusual events. Comparison of the proposed method to traditional labeled discriminatory classifier (polynomial classifier) shows superior performance even without the laborious step of hand labeling. Results on laser welding image sequences show effectiveness of the algorithm with a small estimated false positive rate of 1.8%.

The work by Stauffer and Grimson [27] represents a well-constructed general site activity analysis system. Many components of this thesis have stemmed from the straightforward but also flexible design of modules comprising Stauffer and Grimson's work. The most novel aspect of this previous work is the adaptive mixture of Gaussians background model. The model allows for very robust foreground segmentation which has been adopted for this thesis application of smoke/fire motion detection. Tracking of the

segmented foreground objects is completed using position predictions from Kalman filtered position measurements. Objects are matched between frames based on position and also size. Features are extracted between frames from the tracked objects, including positions, position changes, and size. The extracted features are inputs to an online Vector Quantization (VQ) scheme. VQ is a technique that allows the feature inputs to be combined into regions representative of multiple feature vectors. These combined feature inputs become what are referred to as code vectors, which when compiled together are known as a codebook. VQ in this application serves to create codebooks for development of a probability-based binary tree representation of scene activity. Once developed, this probability function is then used to classify scene activity with only a single observation required per classification. Unusual activity can also be detected if a low probability from extracted features is calculated. The most extensive testing of the system was for people and vehicles in a parking lot setting, but is claimed to also be used successfully to track fish in a tank, ants on a floor, and remote control cars in a lab setting.

Basharat et al. [28] proposed an algorithm that utilizes GMMs for learning positional and trajectory information on a per-pixel basis within a surveillance scene. EM is used to compute GMM parameters in an online fashion when enough tracking information is collected (position, velocity, and size measurements). Anomalous activity is detected using stored trajectory distribution information, detecting abnormal movement speed or positions compared to a training sequence. A novel background model learning rate adjustment is introduced in the paper as well, which modifies the learning rate of the statistical background model depending on the velocity of the moving object. This concept

(with varying approach) has been adopted for use in this thesis as well due to its effectiveness in keeping slow moving scene objects in the foreground for extended periods of time. Results of testing show success in detecting untrained anomalies in speed and trajectory of tracked objects as well as in improving the background update model using size/velocity tracking features.

2.2 Smoke & Fire Detection

The effectiveness of smoke and fire detection devices has always depended largely on specific application. There are many techniques used to detect smoke and fire, and they all have both strengths and weaknesses. Some primary considerations in choosing the correct device for the application are sensitivity, cost, accuracy/distance capability, response time, and nuisance/false alarm immunity. It is also important to understand that certain detection devices are better suited to detecting certain stages of fire development. There are four basic stages in the development of any fire [29]:

1. **Incipient Stage**
    a. No noticeable smoke, heat, or flame
    b. Infrared and ultraviolet radiation signature first appears
    c. Many sources of noise prevalent, deterring simple detection
2. **Smoldering Stage**
    a. Very little flame/heat
    b. Smoke particles produced and become visible for low energy fires
3. **Flame Stage**
    a. Substantial heat generated from flames
    b. Smoke output decreases slightly
    c. Slow increase in air temperature due to convection of thermal energy

4. **High Heat Stage**
    a. Rapidly spreading flames
    b. Extreme heat
    c. Toxic gases released dependent upon burning material

2.2.1 Photoelectric Smoke Detection

One of the most common forms of smoke detection is the photoelectric spot-type smoke detector. This is a typical residential smoke detector that operates by detecting particles in the air through light scatter detection. In most of these devices, a near infrared (IR) light emitting diode (LED) beam is directed upon a target, and any smoke particles that enter the path of the beam will scatter the light. The scattered light is detected by a silicon photodiode that provides an electrical indication of the presence of smoke. The device sensitivity can be adjusted to specifications compliant with Underwriters Laboratory (UL) Standard 268, and like all spot type detectors, have a maximum coverage of around 900ft^2 (30ft x 30ft). Some strengths of this design are low cost and its simplistic nature. One weakness of this design is that it is prone to false alarms caused by nuisances such as cigarette smoke, welding, and cooking hot oil, etc. Activation time can also be high since it requires that smoke particles actually reach the sensor before they can be detected. Over time this type of design can become overly sensitive due to dirt/dust collecting on the light sensing components [29], [30].

2.2.2 Projected Beam Smoke Detection

Projected beam smoke detectors are a variant of the design of the photoelectric type of smoke detector. Instead of measuring the scattering light due to smoke particles, they

measure the amount of light that is occluded by the particles entering the path of light pointed directly at a photo sensor. Projected beam detectors may use near IR or laser light. They have the benefit of being able to detect over large distances since the beam transmitter and beam receiver can be mounted independently across a large ceiling for example. This capability also reveals some weaknesses of the design. Hot pockets of air that collect in high ceilings or atria will inhibit the flow of rising smoke in the path of the sensor beam. If turbulent air exists in the space it will dilute the smoke particle density and decrease the amount of light that is occluded by the beam. Projected beam systems are also fairly costly in comparison to spot type detectors and are not considered capable of detecting smoke in its early stages [29], [30].

2.2.3 Ionization Smoke Detection

Another common form of smoke detector is the ionization spot-type smoke detector. This is a typical residential smoke detector that operates by detecting the change in ionic current induced by a radioactive source. The radioactive source is usually a small amount of alpha radiation emitting Americium. An ionization chamber is created using the radiation, allowing for current change depending on ion attachment to particles entering the chamber space. This allows for extremely small and even invisible particles in the air to be detected since it does not rely on optics like a photoelectric or projected beam detector. Ionization detectors are considered well suited for high energy flaming fires since they exhibit much less visible smoke compared to a smoldering incipient fire. They are susceptible to false alarms due to their high sensitivity to minute smoke particles but are extremely low cost [29], [30].

2.2.4 Heat Detection

Detection of fire by directly measuring ambient temperature is a simple way of determining the presence of fire. Spot type heat detectors exist along with linear heat detectors. Spot type detectors can detect or alarm to heat in a single location at either a specific fixed temperature or if a regular change in temperature is detected (~15°F/min) as in a "rate-of-rise" unit. Spot type heat detectors work best in a small and unoccupied area susceptible to fires that may spread very quickly [30].

Linear heat detectors operate similarly to spot type heat detectors but they are in a rope/cable form and are sensitive over the entire length that they span. One such type of detector places two wires wrapped around an insulator with an electrical current flowing through them. When the insulator melts, the current rises due to the short and an alarm will sound. This design requires that the burned section be replaced after an alarm occurs. Various other electrical-based designs exist, some utilizing thermistor insulation material, linear thermocouple insulation material, negative temperature coefficient wire coating, and semiconductor heat sensor placements. Fiber optic cable-based heat detectors work by detecting an attenuation of the optical signal on one end of the run of cable after the cladding of the cable melts at a specific temperature. Advanced signal processing can also be used to determine the specific area of fiber optic cable that caused the attenuation. The final type of linear heat detector is based on pneumatics. This type detects an increase in pressure caused by the expansion of heated gas in a length of tubing. All of these types of linear heat detectors have specific applications, most notable being installation in highway tunnels due to their advantageous form factor [31].

A major advantage of heat detectors is the low risk of false alarms caused by nuisances such as with smoke detection devices. Heat detectors will generally only alarm if they become dangerously heated or if there is some sort of tampering to the sensor element. A large disadvantage associated with this technology is that it takes a long time for activation to occur. There is a thermal lag that only allows the alarm to go off during the final stage of fire development unless the sensor happens to be placed in close proximity to the source of the fire.

2.2.5 Optical Flame Detection

Optical flame detection (OFD) operates on the principle of detecting wavelengths of light that are radiated by flaming sources. One method of OFD is to sense ultraviolet (UV) light radiation. The UV spectrum is filtered to detect only the wavelength emitted by flames (190-260nm range) and to reject sunlight, incandescent, and fluorescent sources of UV light. The optical flame detector senses light in a conical field of view similar to a camera (~110° horizontally and vertically), which may be a great distance away from the sensor (~50ft depending on mounting height). It will detect the strongest source of energy when the path of radiation is most directly targeted to the sensor, but can also detect indirect reflections from exposed walls and objects. Only detecting the presence of UV energy is not enough to avoid nuisance alarms for sources such as electric welders. To mitigate this problem, the rate of flame flicker is also detected and checked to be within the range of 5-30Hz [30].

OFD may also be achieved by measuring IR radiation, filtered to the 3800-4300nm portion of the electromagnetic spectrum. IR radiation detectors work well in detecting hot

carbon dioxide radiation and like UV detectors, they must detect flame flicker in order to avoid nuisance alarms. The IR optical flame detector has a range that is about half of the UV detector. Hybrid OFD devices containing both UV and IR radiation detection capabilities controlled by microprocessor circuitry allow for the best of both methods. They are considered one of the most advanced fire detection devices available for commercial applications, and are most popular for use in large high risk fire areas such as hangars, chemical production, gas turbines, highway tunnels, and power generation facilities [29-31].

Benefits of OFD devices include fast response time (< 1s), high sensitivity to most types of fires, they can be used indoors and outdoors (unlike spot type detectors), and they are considered very reliable up to relatively large distances. Potential disadvantages of OFD devices are decreased sensitivity from smoke obscuration (for UV only) and also high purchase cost [30].

2.2.6 Video Image Detection Systems

One of the most recent and active areas of smoke and fire detection technology has been the development of Video Image Detection Systems (VIDS). VIDS are real-time image processing based approaches to detecting specific anomalies in standard Closed Circuit Television (CCTV) video streams. They allow for the automatic detection of objects or events in video without the need for human visual inspection. Changes in images acquired only by a CCD (Charge Couple Device) camera sensor are used for analysis. The analysis occurs in either a dedicated processing unit for multiple cameras or an integrated processing unit within a single camera. In the case of this thesis and most of the prior

works discussed previously in Table 1, the application of VIDS is for smoke and/or fire detection.

2.2.6.1 Capabilities & Advantages

The use of VIDS for smoke and fire detection offers several advantages over existing standard smoke and fire detectors [32]:

1. Proven capability of detecting smoke and smoldering/flaming fires at a rate faster than spot-type detectors;
2. Sensitivity and nuisance alarm immunity are capable of performing as well as the corresponding software algorithms backing the video sensing systems. The software can be constantly upgraded and improved to add new features to an existing system;
3. Even with a limited/obstructed field of view, a video system is capable of providing a physically larger range of coverage than a single spot-type smoke detector, and does not require that the smoke be in the immediate vicinity of the camera in order to detect it;
4. Smoke can be detected in the entire field of view, from floor to ceiling in an enclosed environment, unlike a spot-type smoke detector which resides on the overhead and thus can only sense smoke when it moves close enough to the detector. It is possible that smoke may never actually reach the ceiling before being ventilated in a smaller type of fire, in which case the spot-type detector is ineffective;

5. Ability to use basic components such as cameras and wiring for multiple purposes other than fire detection. The fire detection component can be added to an existing surveillance system via software, minimizing installation, maintenance, and service costs of a replacement system. Standard smoke detectors would need wiring to each sensor throughout an entire ship in all monitored spaces. Cameras would need a single video/data cable to be connected and run to each camera;
6. Maintenance and testing of standard spot-type smoke detection systems requires that each device in the system be checked out rather than a single main computer as in a video detection system. The main computer can monitor all system health parameters from a central location;
7. Ability to have a live video feed of the region where an alarm condition is detected. Instant visual assessment of a problem or nuisance can be done by an operator. This is in contrast to a standard fire alarm system which gives only an alarm notification of a smoke condition. The scene of the alarm must be checked out in person before the situation can be assessed, wasting valuable time in the event of a real danger;
8. The video detection system can be used for more than just smoke/fire detection with algorithm developments. Additional uses might be personnel tracking, activity recognition, flood detection, and physical damage assessment aside from basic surveillance which is inherently integrated into VIDS.

2.2.6.2 VIDS Standards & Regulations

At the time of this writing, a formal set of standards tailored specifically to VIDS has not yet been developed by all major standardizing bodies such as the National Fire Protection Association (NFPA), Factory Mutual (FM), and Underwriters Laboratories (UL). The NFPA 72: National Fire Alarm and Signaling code "covers the application, installation, location, performance, inspection, testing, and maintenance of fire alarm systems, supervising station alarm systems, public emergency alarm reporting systems, fire warning equipment and emergency communications systems (ECS), and their components" [33]. The NFPA 72 code is updated every 3 years and the most current edition is the 2010 edition. Although the code refers to the use of video sensing systems, it currently does not address the requirements of such systems. Only the manufacturer's published instructions can be used as a performance-based measurement.

FM (Factory Mutual) standard 3260 for Radiant Energy-Sensing Fire Detectors for Automatic Fire Alarm Signaling applies to VIDS, although it was originally meant for optical flame detectors. It states that manufacturer-specified sensitivities are to be used to detect one or more of specified sizes of n-Heptane, alcohol, JP4 jet fuel, or propane flame fires [34].

The Underwriters Laboratories (UL) has just recently developed a specific standard that refers to VIDS. UL 268b states that "the Video Image Smoke Detectors addressed by this outline are to be investigated with the intent to detect the image of smoke from a fire in the field of view area defined by the limits of the camera. Installation/operating document(s) and/or product marking information provided with the product must

describe the specific operating parameters of the product. This information assists in defining some of the limits required for the certification program." It also states that "although Video Image Smoke Detectors may include flame detection, this outline does not include testing for flame detectors or the combination of Video Image and Flame detection" [35]. This is currently the only standard specifically developed for VIDS, but only applies to systems that detect smoke and manufacturer specifications are referred to just as per the FM 3260 standard. The outline of the standard shows that the testing is quite exhaustive and includes many test aspects such as sensitivity, directionality, velocity sensitivity, reduced lighting, smoldering smoke tests, varying types of fire tests, electrical fire tests, replacement, durability, survivability, and endurance [36].

The reason for specific standards being in limited development is due mainly to the early age of the technology. There are also a great number of variables that need to be addressed compared to other fire detection methods. It is particularly difficult to make blanket statements about all forms of VIDS for smoke and fire detection because they use unique and proprietary algorithms/hardware to perform their functions. Additionally, these functions are not identical between the systems. Some systems may only detect smoke rather than fire, and the detection times and distances vary. It is not yet clear what the minimum functionality of VIDS should be, therefore it is up to the end user to decide what suits their application the best.

2.2.6.3 Commercial VIDS Implementations

Recently there have been an increasing number of commercial off-the-shelf (COTS) implementations of VIDS technologies developed around the world. They are marketed

under unique product names with no standard terminology for describing the systems, but their functions are generally the same. They detect the presence of smoke and/or fire as quickly as possible and relay the information to a main software monitoring station. They may also be integrated into a standard fire alarm panel to trigger a building alarm.

The AxonX Signifire IP system has the distinction of being the first video smoke detection system to gain the UL 268b approval. AxonX is also the only company to release any insight regarding the actual algorithms used for processing video streams in their product. This is in an effort to provide a more open discussion of the technology that will lead to a better quality and expedited development of NFPA and similar standards. The Signifire system is capable of smoke and flame detection. The flame detection aspect works by detecting regions with slow changes in brightness and also changes in the edges of those regions. Morphological processing groups the fire regions together and normalizes the regions to a standard size of 7x7 pixels. A neural network is then used to distinguish the normalized object as fire or a nuisance source. The neural network has been trained with over 5,000 flame samples and over 10,000 nuisance source samples. The fire source samples are representative of diverse data including extreme cases during wind. Nuisance source samples come from possible light sources, reflections, strobes, computer monitors, and human/vehicle traffic [37].

The Signifire system also employs a smoke detection algorithm. The algorithm analyzes monochromatic video by detecting intensity changes and updating a pixel change history such as in [20]. The regions of most change are then extracted and normalized to 7x7 pixels and classified by a neural network. The neural network is

different from the flame detection network but trained similarly with over 15,000 samples of smoke in varying scenarios as well as nuisance sources like moving people/objects and lighting changes [37].

The Fastcom Smoke & Fire Alert (SFA) system uses contrast, color, movement, outline, and other unspecified information from up to 16 CCTV cameras to detect smoke and fire on a central processing PC. The Micropack FDS-301 Image Based Flame Detector, Fire Sentry VSD-8, and the D-Tec FireVu systems are three more examples of COTS VIDS implementations that have similar capabilities but do not list specific methods or insights into the underlying algorithms used for smoke and fire detection.

2.2.7 Multi-Sensor Fire Detection

Recently there has been research and development of prototype multi-sensor systems that package a variety of fire and smoke sensors into a single device. The Early Warning Fire Detection (EWFD) system developed by the Naval Research Laboratory (NRL) is a multi-sensor system that utilized many combinations of gas, humidity, and temperature sensors as well as standard smoke detectors. It combined the outputs of these sensors using pattern recognition techniques with a probabilistic neural network (PNN) as the method of classification. The EWFD was successful and had "better overall performance than the commercial smoke detectors used without the algorithm, by providing both improved nuisance source immunity with generally equivalent, or faster response times" [38]. Some drawbacks of the EWFD were that the detection of smoldering fires needed improvement and detection times could be decreased if fire/smoke particles did not need

to actually reach the sensor array to permit detection. This led to the creation of the advanced volume sensor project.

Also developed by the NRL, the volume sensor is another example of a multi-sensor fire detection device. It was called a volume sensor due to the diverse array of sensors used to monitor a volume or space. The device integrates IR, UV, CCD, VIDS, and acoustic sensors into one unit, managed by software algorithms that perform multi-sensor data fusion to distinguish the presence of smoke, fire, explosions, or pipe ruptures from normal nuisance events in a shipboard environment. Some particular novelties of the system are the utilization of "long wavelength" imaging, or night vision imaging to detect infrared characteristics of fire spatially rather than spectrally. Microphones are used to detect loud noises, particularly of pipe bursting and human nuisance events such as welding, grinding, conversing, etc. Most importantly, the volume sensor does not rely on smoke particles or heat coming in close proximity to the sensor in order to detect smoke or fire.

Video image analysis, acoustic analysis, long wavelength video detection, and spectral sensor analysis have all been integrated into a multi-sensor data fusion method of classification for the final volume sensor prototype. The resulting overall system is capable of monitoring events in real time, providing pre-alarm and alarm conditions for unusual events, logging and archiving data from each subsystem, and archiving and indexing alarms for future recall. It is a modular design that allows for future expansion into developing sensor technologies and for separate applications to be added later. The system works by first processing raw data using algorithms at each subsystem level. The

data is then combined and analyzed across all of the subsystems using decision tree pattern recognition techniques for pre-alarm condition awareness. A pre-alarm triggers a second tier of more advanced pattern recognition analysis involving feature selection, clustering, Bayesian classification, Fisher discriminant analysis, and neural network algorithms [39].

The two volume sensor prototypes that were built at the time the report was released consist of a sensor suite including a video camera, long wavelength filtered video camera, spectral sensors, and a microphone. A computer analyzes the data at each subsystem level and packages it to be passed onto a single fusion machine for event recognition. The prototypes are similar, but vary in the type of VIDS employed. The VIDS used by the first prototype is the Smoke and Fire Alert software package from Fastcom Technology and the second prototype used the Signifire fire detection software by AxonX. Both VID systems were customized by the manufacturers to interface properly with the volume sensor system. They also have a main graphical user interface (GUI) to display in real-time what alarm conditions may exist, as well as sensor states and a view of three different cameras that are selectable to be any of the cameras in the system [39].

Investigative studies performed prior to development of the sensors [40] proved that new technology such as VIDS were advantageous over existing spot type smoke detectors, as well as which existing systems exhibited the best performance [37]. Experimental testing was completed for initial VIDS performance testing and the results provided evidence that the VIDS work well in a simulated confined shipboard environment and work even better than traditional smoke detectors overall. The results

gave good reason to have the systems tested in a real shipboard environment [41]. Shipboard environment testing was performed after the development of the volume sensor prototype aboard the ex-USS *Shadwell* Navy training platform in Mobile Bay, Alabama. There were three basic metrics used for determining how successful the system was during the tests. The first metric was the overall proper functioning of the system as it was designed to perform. The second metric was the number of correct classifications vs. false alarms for detecting sources of fire, water, gas releases, and nuisance sources. Lastly, the speed of the system's response to detecting fire sources was measured. The effectiveness of the prototypes was also compared to stand-alone COTS VIDS and spot-type ionization and photoelectric smoke detectors. The sensor data and analysis data were transmitted to a main data fusion PC once per second [40], [42].

The results of testing the system showed that the visual detection methods (VID and long wavelength) failed at reliably discriminating bright light nuisance sources such as welding, cutting, and grinding from flaming fires. The spectral sensors were useful for this cause, because they could use their already incorporated welding and fire sensing capabilities to provide complementary information able to be incorporated into the data fusion pattern recognition algorithms. VIDS detected all smoldering fires, while the spectral sensors and long wavelength system detected about half of the smoldering fires but nearly always at a rate slower than the VIDS. The volume sensor prototype as a whole depended most on the VIDS to detect smoke/smoldering. However, most of the false alarms were also caused by the VIDS subsystem. The acoustic subsystem performed well

for pipe rupture and gas leak events, and also fairly well for nuisance sources, despite the fact that the acoustic nuisance detection was not quite mature at the time of testing [39].

The volume sensor prototypes were able to detect all sources of flaming and smoldering fires. The correct classification of nuisance sources resulted in 72% for the first prototype and 78% for the second prototype, with the incorrect classifications corresponding to false positives. The prototype systems were better overall at rejecting nuisance sources as alarm conditions with the best detection of smoke and fire events. The pipe rupture tests resulted in 94% correct classification for the prototypes, with the only failure occurring during a weak flow rate test. This was far better than the commercial VIDS due to additional acoustic information that was integrated. It was also found that the prototype systems provided response times that were faster or equal to the other VID and spot-type systems tested, i.e. within 30 seconds or less [39].

2.3 Shipboard Damage Control Systems

Damage control (DC) systems aboard a Navy ship are responsible for providing the capability of the ship to take on damage and continue operating towards completing its mission [42]. This continued operation is known as the ship's survivability. Survivability deals with these important aspects:

- **Susceptibility** – "The degree that the ship is open to attack," either due to battle, equipment failure, weather, or accident;
- **Vulnerability** – "The likelihood that the ship would be lost if the attack was successful and the ship was hit;"

- **Recoverability** – "The ability of the ship and its crew to survive an attack and maintain/restore capabilities essential to the mission."

It is also useful to discuss the eight areas that the US Navy defines to be of most importance in shipboard DC [42]:

1. **Identification and Assessment** – Capability of rapidly determining when damage has occurred and the severity of the damage either during battle or otherwise;
2. **Communication** – Effectiveness of relaying messages and information within a ship in a timely manner;
3. **Management** – The way that the DC system components are organized and deployed, either automated, semi-automated, or manually;
4. **Action** – Capability of the damage control system to act upon the situation based on the state of the ship, either automated, semi-automated, or manually;
5. **Personnel Training** – Capability of training personnel to properly utilize damage control system components to maximize ship survivability;
6. **Logistics** – Staying aware of limitations on a system wide level for integration of new components in terms of cost and technical feasibility;
7. **Maintenance of Damage Control Equipment** – Capability for damage control components to have minimal upkeep and self-diagnostics/calibration in a way that is as cost and time efficient as possible;
8. **Development and Assessment of Next Generation Damage Control Tactics/Equipment** – Future cost and effectiveness relies directly on the improvement of existing methods that must constantly be researched.

2.3.1 Fire Suppression Methods

The suppression of fires once they have been detected on a ship is considered part of the action area of DC as defined in the previous section, and may be accomplished in various ways. The suppression capability for a ship depends on the type and generation of ship,

since all vessels have different needs and specific dangers associated with them [43]. Aside from shipboard firefighting and standard sprinkler systems, there are a few basic methods used that will be discussed as well as the direction of future technology in this area.

2.3.1.1 Gaseous Agents

The use of gaseous agents for fire suppression is based on the concept of flooding a space with inert gas that displaces oxygen and effectively removes oxygen from the combustion reaction. The most popular gases used for this purpose have been a type known as halons. "Halons are a class of halogenated hydrocarbons that are highly effective in suppressing combustion and that, accordingly, are widely deployed on the ships and aircraft of the US Navy." Specifically, halon refers to halon 1301 (CF_3Br, commonly used) and halon 1211 (CF_2ClBr, limited use). Halon is desirable for many reasons including ease of distribution in obstructed spaces, low toxicity, and storage stability. Unfortunately, the production of halons has been banned by international treaty and US law due to studies determining that the gas is ozone-depleting. However, the military was granted permission to use all existing supplies of halons since a replacement is not currently available. There has been research to develop a drop-in replacement for halons in the existing systems that has no negative environmental effects; however there is also enough supply of existing halons for the life of ships currently using it. A type of gaseous agent requiring only minor hardware modifications is HFC-227ea. It requires more than two times the weight and storage volume of halons, but offers a solution to the problem of halon replacement if the Navy's supply is depleted [44].

Carbon dioxide (CO_2) is also used as a fire suppression system in some shipboard spaces, particularly those that contain flammable liquids or chemicals. "The CO_2 total flooding system is manually actuated except in a few special applications when it is automatically actuated by detection of heat, flame, or smoke. Upon actuation, the pressurized CO_2 storage cylinders discharge through piping and nozzles into the protected space. When discharged into the space, a large portion of the liquid CO_2 flashes to vapor and the rest is converted to fine dry-ice particles" [45].

2.3.1.2 Water Mist

The use of water mist fire suppression technology is seen as one of the best alternatives to existing halon systems in Navy ships. The concept of water mist suppression is quite similar to a standard sprinkler system except that the water is dispersed at a much greater pressure (~1000 PSI). The greater pressure allows for smaller particles to be dispersed, occupying a larger volume of air and using much less water than a sprinkler system. The mist serves the purpose of cooling flames, wetting surfaces, and at the same time displacing oxygen from steam expansion similarly to a gaseous agent [44]. Water mist requires high pressure and fresh water; therefore it is deployed using stored water fed through a high pressure pump [42]. The advantages of water mist suppression include low cost, no toxicity, no adverse environmental effects, suppression of flammable liquid pool and spray fires, and explosion suppression. In order to retrofit existing halon systems with water mist systems, it would require significant modifications that may not be feasible. Two pump rooms are required, along with hundreds of feet of pipe, so it is more sensible to use a halon replacement such as HFC-227ea [44]. Water mist systems are also limited in

their ability to be effective in mission critical environments such as engine rooms, according to thorough testing [46].

2.3.1.3 Aqueous Film Forming Foam

Aqueous Film Forming Foam (AFFF) systems are used mainly in ships for hanger bays, machinery spaces, and bilge deluge systems [42]. Certain types of AFFF that are no longer in use had environmental concerns due to the way that they broke down over time. AFFF is still in use and testing has proven its effectiveness in the treatment of small to moderate sized fires. AFFF may be used as part of a High Expansion Foam Fire Suppression System (HEFFSS) as well, which uses a motorized foam generator and a foam proportioning system.

2.3.1.4 Aerosols

Aerosols are a class of fire suppression agents that work by dispersing particulates in air. The small particles do not suffer from the possible negative environmental effects of gaseous agents. They inhibit the chemical reactions in fires and cool flames most effectively in larger shipboard spaces. The downside to aerosols is their tendency to reduce or completely obscure visibility in confined spaces. Also, some types of aerosols are generated by pyrotechnic reactions which make them very hot and potentially dangerous to personnel [42].

2.3.1.5 Smart Valves

Smart valves refer to automated valves that contain sensors and processing capability that allow them to have situational awareness. Smart valves contain a valve, actuator,

microprocessor, communication transceiver, pressure sensors, and control logic for remote control. This gives the valves functionality to divert or cutoff the flow of water to areas that are in specific need or have been damaged. In the event of a fire, a smart valve may divert additional water to fire suppression systems. The rupture of a pipe may trigger the valve to cut off water flow in that section. The use of smart valves gives the DC system a localized action without the need for the entire management system to be aware of the fault [42].

2.3.1.6 Personnel and Equipment

Besides the use of large scale fire suppression systems, there exists a large variety of equipment for personnel. Ships are equipped with portable smoke equipment, portable fire extinguishers, oxygen breathing apparatuses, self-contained breathing apparatuses, portable dewatering equipment, forcible entry tools, and also firefighting helmets, boots, gloves, and flash gear [47]. The use of Navy personnel could be considered its own form of fire suppression since they may ultimately make the difference in maximizing ship recoverability. Firefighting on a ship can be a very difficult task, since most fires occur at engine room compartments where hot flammable liquids are present. Firefighters must travel through spaces filled with superheated gases to even get a glance at the problem [48].

With an established fire below decks, Navy firefighters' tactic is to lock down the affected area. "They stifle its air supply by shutting down ventilation systems and depriving the fire of its ability to extend by establishing boundaries on as many sides of the space involved as possible. Boundaries can be set at two levels. Primary boundaries

would use the decks and bulkheads forming the compartment that is on fire, boxing it in. Secondary boundaries are established by sealing off the vessel's watertight bulkheads and deck openings surrounding the compartment on fire, creating a larger box. Once the boundaries are established, they are maintained by cooling the decks and bulkheads with water and removing exposed combustible materials. Withholding ventilation until boundaries are set may well be the appropriate action to slow the spread of the fire. Only after this is done would a ship's crew turn its focus to extinguishment" [49].

Aside from firefighting, specific personnel duties also may involve preventing fires. Fire watch duties are given whenever hot work such as welding, cutting, and brazing is done. The fire watch must watch the affected area for up to 30 minutes after the work is completed [50].

2.3.2 Naval Damage Control Facilities

Nearly all of discussed technologies as well as the previously mentioned NRL volume sensor have been thoroughly tested aboard Navy damage control facilities. The largest of these facilities is the ex-*USS Shadwell*. Located in Mobile Bay, Alabama, it is an LSD-15 ship built in 1944 and decommissioned in 1970. It has been converted into a full-scale DC test facility "to give an integrated picture of the interactions of man, equipment, materials, doctrine and systems in a realistic shipboard damage environment. The ex-*USS Shadwell* serves as the ultimate test platform in the development of firefighting agents, DC systems, predictive models and technology stemming from basic and theoretical concepts developed through naval research. The ex-*USS Shadwell* also serves as realistic shipboard test platform for endeavors other than DC that evolve from research and development in

other disciplines, such as coatings, insulation, working fluids, cleaners and communications" [51].

The Chesapeake Bay Fire Test Detachment (CBD) is another test facility that "is concerned with all aspects of shipboard fire safety, particularly as related to flight decks, submarines and interior ship conflagrations. The emphasis is on providing facilities for intermediate scale, credible evaluations of firefighting agents, systems and training concepts under more realistic, shipboard conditions. In many cases, the facility provides a vital link between laboratory testing and full scale, proof of concept on the ex-*USS Shadwell*" [52].

2.4 Foreground Segmentation Techniques

In order for the video sensing system to be useful, it must be capable of operating in real-time, with a suitable frame rate for surveillance analysis. Additionally, it must be robust and able to be employed in various operating conditions, with unknown exact lighting or background. With these conditions in mind, a suitable foreground segmentation technique was sought for the system using a single, stationary, monochromatic surveillance camera.

Foreground segmentation is the separation of foreground pixels from background pixels in an image sequence. Foreground pixels are those which belong to any moving objects, and represent a significant change in a pixel's value from the previous frame. The result of this segmentation is a binary image in which white pixels represent the foreground objects in a scene, and black pixels represent the background. There are many

different ways of performing foreground segmentation, and they all have specific strengths and weaknesses. The methods of temporal differencing, optical flow, and statistical background modeling will be discussed in the following sections.

2.4.1 Temporal Differencing

One of the most common methods of foreground segmentation involves a simple subtraction of the previous frame from the next most recent one in a sequence. The difference is then converted to a binary representation based on a set threshold value. This method of temporal differencing is an effective way of capturing every amount of change in the frame, but it is highly sensitive to noise in the acquired camera images, changes in illumination, and cannot account for complex backgrounds. A slight adjustment to this method involves using a moving average to model the past n frames as the background of the sequence. This provides better results than a 2-frame difference, but results in a trade-off between learning the foreground too quickly and detecting false positive foreground objects. Further improvements of this method use a learning rate parameter to allow for a weighted moving average to represent the background, giving favor to the less recently seen images [53].

$$B_{i+1} = \alpha \times F_i + (1 - \alpha) \times B_i, \tag{1}$$

where α is the learning rate, typically around 0.05, with B as the modeled background, and F as the sequence image. The foremost benefit of temporal differencing methods are their speed and simplicity; however this simplicity is at the expense of precision. Also, temporal differencing does not allow foreground regions to be persistently tracked over time since only instantaneous motion can be determined.

2.4.2 Optical Flow

Optical flow can be defined generally as "the distribution of apparent velocities of brightness patterns in an image," or more specifically as "the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and the scene" [54], [55]. It has a large variety of applications, including motion detection and segmentation, video compression, modeling three-dimensional scene structure, scene change detection, robot navigation, and cancelling out uniform background motion in a sequence for a non-stationary camera such as in UAV (Unmanned Aerial Vehicle) application.

Optical flow can be estimated from a variety of different methods, some involving differentials, region-based matching, energy-based, phase-based, or feature-based techniques [56]. The more popular of these methods seems to be the spatiotemporal derivatives-based methods such as the one developed by Horn and Schunck [54]. Regardless of how the flow vectors are estimated, the general process is to apply a combination of low/high pass filters to smooth the image and enhance the signal to noise ratio, perform the calculations to measure normal velocity components, and then finally integrate the results, producing a 2D flow field/image velocity field (Figure 1) [54]. The flow field shows how regions of an image move over time, usually separated into a rectangular grid, since neighboring pixels generally move in the same manner.

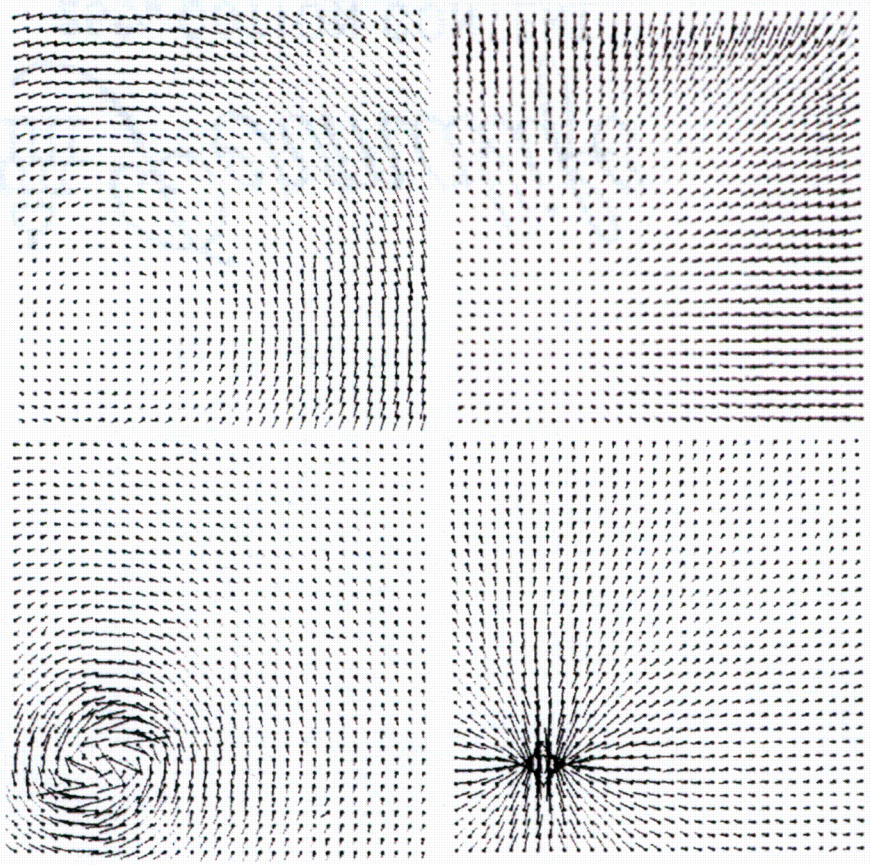

Figure 1. Image velocity dense flow field resulting from rotation (top left), contraction (top right), vortex (bottom left), and a sink (bottom right).

Some additional focus in this section is on the method of optical flow estimation first introduced by Horn and Schunck [54], which is still one of the most popular techniques in light of alternatives that have been presented since it was published in 1981. The initial portion of the method's derivation will be discussed as it is relevant to nearly all methods of optical flow estimation. The method utilizes partial derivatives of image pixel intensities in the x and y directions over time. The accurate estimation of optical flow parameters relies on the basic fact that the intensity of pixels corresponding to a moving object in the scene of video or moving image sequence is constant, at least for short periods of time (Figure 2) [57]. This means that the illumination in a scene should

be constant and not variable as a function of position. There should be no flickering or dimming lights, or shadows in the path of the moving object. These conditions are of course for an ideal situation, which is why much of the testing that is done on optical flow methods are carried out using artificial sequences created from three dimensional object models with known optical flow prior to testing. This is in contrast to real-world sequences where the optical flow can only be estimated, and it is not known for sure as to what to place a benchmark upon the tested method of calculation.

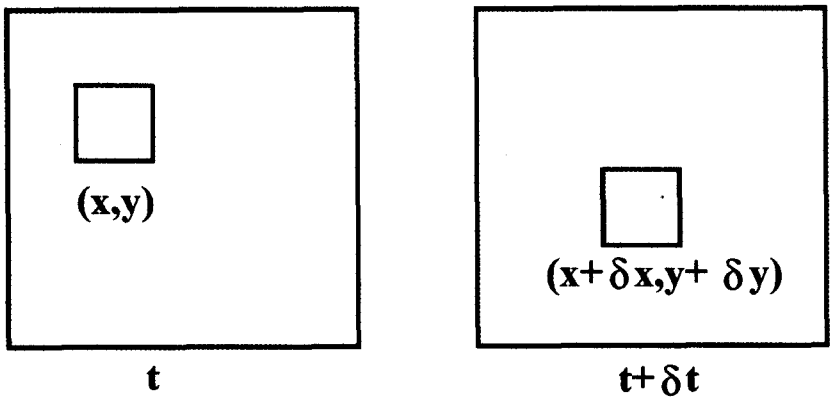

Figure 2. Illustration of motion constraint equation where intensity is constant over small periods of time.

We first define the image intensity at a point (x, y) in an image at time t as a function $I(x, y, t)$. Since we are assuming that the intensity at a point is constant, it is not changing and therefore,

$$\frac{\mathrm{d}I}{\mathrm{d}t} = 0. \tag{2}$$

If we consider a region that has moved in some direction over a period of time, and the intensity remaining constant (Figure 2),

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t). \tag{3}$$

This can be expanded to

$$I(x,y,t) = I(x,y,t) + \delta x\frac{\partial I}{\partial x} + \delta y\frac{\partial I}{\partial y} + \delta t\frac{\partial I}{\partial t} + \varepsilon, \tag{4}$$

where ε includes the higher order terms in $\delta x, \delta y,$ and δt for a Taylor series expansion about the point (x,y,t) [58]. Subtracting both sides by $I(x,y,t)$ and dividing by δt,

$$\frac{\delta x}{\delta t}\frac{\partial I}{\partial x} + \frac{\delta y}{\delta t}\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} + \mathcal{O}(\delta t) = 0. \tag{5}$$

Where $\mathcal{O}(\delta t)$ includes the higher order terms which approach zero as δt goes to zero which yields,

$$\frac{\delta x}{\delta t}\frac{\partial I}{\partial x} + \frac{\delta y}{\delta t}\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0. \tag{6}$$

Rewriting $\frac{\delta x}{\delta t}$ and $\frac{\delta y}{\delta t}$ as the components of image velocity in the x and y directions as v_x and v_y,

$$\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t} = 0. \tag{7}$$

It is important to note that v_x and v_y represent the optical flow velocities of the region, and $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$, and $\frac{\partial I}{\partial t}$ represent the rate of change of intensity in the region in the x and y directions and over time. This can be rewritten using an alternate notation,

$$I_x v_x + I_y v_y + I_t = 0, \tag{8}$$

$$(I_x, I_y) \cdot (v_x, v_y) = -I_t, \tag{9}$$

$$\nabla I \cdot \vec{v} = -I_t, \tag{10}$$

where ∇I is the spatial intensity gradient and $\vec{v}$ is the image velocity of a point (x,y) at time t. This is known as the 2D motion constraint equation. It is an equation with two

unknowns; a mathematical representation of what is known as the aperture problem. The aperture problem occurs when the direction of motion cannot be fully determined when looking through a small aperture. As an example, refer to Figure 3(a) [59]. The line is moving in the direction of $\vec{v}$, however only the velocity of the normal component of the intensity gradient, $\vec{v}_n$ can be determined. Figure 3(b) shows the result of the motion constraint equation in $\vec{v} = (v_x, v_y)$ space, where the line represents all possible velocity vectors from the given problem, but only one of the points on the line is the correct one. The normal velocity component, $\vec{v}_n$, is the vector with the smallest magnitude from the origin on the line.

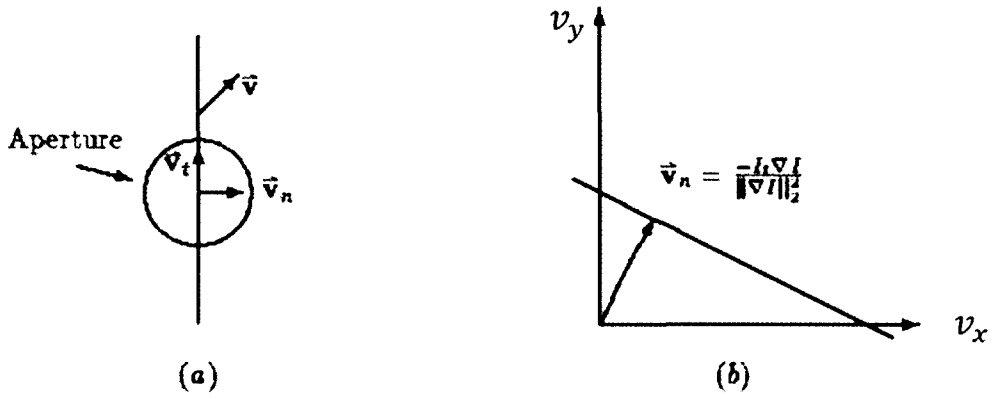

Figure 3. (a) Illustration of aperture problem for moving line viewed through circular aperture (b) Line representing all of the possible image velocities in the x and y directions with normal velocity shown.

The rest of the derivation will not be shown in this thesis, but a final solution is determined using additional assumptions based on smoothness in neighboring regions. According to Horn and Schunck, the optical flow can only be determined at the areas in an image with a gradient. The "smooth" areas with little contrast and no edges will eventually follow the estimates of local or surrounding edges. The estimate for these

regions will get better as the number of iterations increases. This occurs because the averaged optical flow estimates propagate inward from the image edges over each iteration. In practice, images may be very large requiring hundreds of thousands of calculations per second in a real-time application, therefore the iterations will be kept to the least number necessary for required accuracy of estimate for the application. "The number of iterations should be larger than the number of picture cells across the largest region that must be filled in" [54].

The work in optical flow was popular in the 1980's and 1990's time period due to computer vision becoming a possibility in terms of computational complexity and hardware availability. Digital image processing methods for implementing optical flow methods were attempted by many researchers. The differential methods of Horn and Schunck [54] and Lucas and Kanade [60] are still considered to be fairly good methods of determining optical flow in terms of accuracy and more importantly, ease of implementation. Source code for these methods can be obtained rather easily due to the fact that they have been around since 1981. Other methods have appeared that claim to have better performance and/or accuracy, however only marginally so, and the source code is not available [61].

Some of the more popular methods for calculating optical flow estimates other than those mentioned include the following [56]: Fleet & Jepson [62] and Waxman, Wu, & Bergholm [63] used a phase-based approach, Anandan [64] used a Laplacian pyramid approach to region–based matching, Singh [65] used a two-stage matching method, and Heeger [66] used an energy-based velocity tuned filtering approach. Liu et al. [67]

developed a method which calculates optical flow "under perspective projection to derive an image motion equation that describes the spatiotemporal relation of gray-scale intensity in an image sequence, thus making the utilization of 3-D filtering possible." A correlation-based approach by Camus [68] reduces computational complexity of a search methodology by completing the search linearly over time rather than quadratic locally.

Much of the recent works in the area of optical flow estimation are actually methods of improving estimates of existing methods, or attempting to speed up the calculation of the optical flow estimates. Bodily [57] implemented GPU-based calculation methods on existing algorithms as well as FPGA-based implementations to determine validity of the techniques and gains in speed over software implementations. Simoncelli [69], Comaniciu [70], [71], Haralick [72], and Marik [73] all dealt with the improvement of stability and accuracy of estimates of existing optical flow estimation methods using statistical methods.

Optical flow is a very powerful technique; however it is at the expense of computational complexity. Additionally, the most well-established techniques based on partial derivatives can only be accurately calculated at regions of high spectral frequency (image edges). It is very sensitive to camera noise, and also it does not allow foreground regions to be persistently tracked over time since only instantaneous motion can be determined. The combination of these weaknesses makes it unsuitable for real-time video streams without specialized hardware [74], [56].

2.4.3 Statistical Background Modeling

Effective foreground segmentation algorithms rely on the principle of subtracting the learned background of an image sequence from the most current image. This method relies heavily on an accurate representation of the background in the image sequence. Maintaining a statistical distribution of the previously seen images is the best way of achieving this. Statistical methods create a way for the background in a sequence of images to be learned to an arbitrary level of complexity, so that even multi-modal backgrounds can be tolerated without being falsely detected as foreground. This is done on a per-pixel level, so that intensities of neighboring pixels are independent of one another.

The way to programmatically learn the distribution of each pixel over time becomes a problem of density estimation. The difficult aspect of this desired operation is that in order to maintain the most accurate depiction of the background, all previous intensities, or a recent subset that is sufficiently large, would need to be maintained in memory for each individual pixel. In an image sequence of only 5 seconds long at 5 FPS, and 200 x 200px in size, the result would be a total accumulation of approximately one million data points to process. In a real-time application, this number would only increase over time. A sacrifice in retained data window width would be required, or more appropriately, an unsupervised online estimate of the distribution could be employed.

2.4.3.1 Adaptive Gaussian Mixture Models

The statistical density estimation technique used in this implementation is that of a mixture of Gaussians approach. As a progression from the unimodal single Gaussian

background model described in the Pfinder algorithm [75], this method attempts to model the background of an image sequence using a mixture of K Gaussian densities at each pixel. This representation is given by

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathrm{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{11}$$

where the mixing coefficients, π_k, valued between 0 and 1, represent $p(k)$, the prior probability or responsibility that the k^{th} Gaussian represents the data point, $\boldsymbol{x}$, given $\mathrm{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, the multivariate normal probability density function (PDF):

$$\mathrm{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathrm{k}}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right), \tag{12}$$

with mean, $\boldsymbol{\mu}_k$, and covariance, $\boldsymbol{\Sigma}_k$. This is equivalent to $p(\boldsymbol{x}|k)$, the likelihood or probability of $\boldsymbol{x}$ given k. $p(\boldsymbol{x}|k)$ represents the probability of observing the data vector, $\boldsymbol{x}$, given that $\boldsymbol{x}$ comes from the kth Gaussian component having mean, $\boldsymbol{\mu}_k$, and covariance, $\boldsymbol{\Sigma}_k$. The posterior probabilities are then given by Bayes' rule as

$$p(k|\boldsymbol{x}) = \frac{p(k)p(\boldsymbol{x}|k)}{\sum_{l=1}^{K} p(l)p(\boldsymbol{x}|l)} = \frac{\pi_k \mathrm{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{K} \pi_l \mathrm{N}(\boldsymbol{x}|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}, \tag{13}$$

which represents the responsibility that the kth component takes for explaining the data vector, $\boldsymbol{x}$.

Deviating briefly from the implemented algorithm, the method of solving for an exact solution for a GMM requires the use of the EM algorithm. In the case of a single Gaussian density, maximum likelihood estimation (MLE) is used to determine the parameters that best fit the data. These parameters are said to be fixed, but unknown, and MLE seeks to solve for these parameters by maximizing the log-likelihood of the

distribution. In the case of multiple Gaussian densities, this problem becomes more difficult because a standard maximization of the log-likelihood will never converge on a closed solution due to the presence of the additional summation inside the likelihood function:

$$\ln[p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] = \sum_{n=1}^{N} \ln\left\{\sum_{k=1}^{K} \pi_k \mathrm{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right\}, \tag{14}$$

where $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_k\}$, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k\}$, and $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k\}$. Also present is a latent variable, z_{nk}, which represents the label for the Gaussian (1-of-K) that best fits the nth data vector, $\boldsymbol{x}_n$. In other words, each data vector has a z_k latent variable associated with it, with $z_k \in \{0,1\}$ and $\sum_{k=1}^{K} z_k = 1$. To solve this problem, EM is a general algorithm that can be adapted for use in GMM density estimation. A thorough explanation can be found in [76], but a general description is given in this section. EM for GMM is an iterative algorithm that begins by initializing the π_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$ parameters and calculating the first log-likelihood estimate using (14). In the E-step, the expected value of the latent variable, z_{nk}, is calculated by

$$\gamma(z_{nk}) = \frac{\pi_k \mathrm{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathrm{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}, \tag{15}$$

where $\gamma(z_{nk}) = p(z_k = 1|\boldsymbol{x}_n)$, the conditional probability of z given $\boldsymbol{x}$ for the nth data vector. $\gamma(z_{nk})$ represents the posterior probability or responsibility that the kth Gaussian component takes for explaining the nth data vector, $\boldsymbol{x}_n$.

The E-step is then followed by the M-step, in which the values for π_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$ are updated by maximizing each with respect to the log-likelihood function by

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{nk})\, \boldsymbol{x}_n, \tag{16}$$

$$\hat{\boldsymbol{\Sigma}}_k = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{nk})\, (\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_k)^{\mathrm{T}}, \tag{17}$$

$$\hat{\pi}_k = \frac{N_k}{N}, \tag{18}$$

where $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$. The log-likelihood is then calculated using the updated GMM parameters, $\hat{\boldsymbol{\mu}}$, $\hat{\boldsymbol{\Sigma}}$, and $\hat{\pi}$ in (14) and checked for convergence based on a minimum threshold of change from the previous iteration. If convergence does not occur, this process is repeated from the E-step in an additional iteration. Due to the monotonically increasing nature of the log-likelihood function, maximizing will always result in convergence.

EM was ultimately not implemented due to complexity for real-time application and also requiring a large window of data for a proper background estimate, but it is useful in discussion of the proper method that a GMM is solved for a data set, and understanding the required approximation. The solution implemented in this application is a method described by Stauffer and Grimson [27], [77], [78] that provides a proper, unsupervised, online approximation of the GMM densities for each pixel in an image sequence. The algorithm may be exercised on RGB color video or monochromatic video. The algorithm they described, works by creating K Gaussian distributions, each with a weight, ω_k, mean, $\boldsymbol{\mu}_k$, and covariance, $\boldsymbol{\Sigma}_k$, approximated by $\boldsymbol{\Sigma}_k = \sigma_k^2\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. In the case of monochromatic video stream, μ_k is scalar as well as Σ_k

which becomes equivalent to σ_k^2. K must be predetermined for the application, but usually 3 to 5 is sufficient.

The parameters for each distribution are first initialized ($\omega_k = 0.05,\ \mu_k = 0,$ $\sigma_k = 30$), chosen by both suggestion of practical values determined by [78] as well as through experimental testing. Then at time t, when a new image in the sequence is acquired, for each of the distributions, the distance between the new pixel intensity value, $\boldsymbol{x}$, and the mean, $\boldsymbol{\mu}_k$, is calculated. If the calculated distance is less than a specified number of standard deviations from the mean (2.5), then the pixel intensity is said to fit into that distribution. The 2.5 standard deviation distance is given in [27], [77] but may be perturbed based on variation in lighting or contrast in the source video stream. Decreasing this distance will allow the algorithm to be more sensitive in detecting subtle background changes. The distribution that fits the data with the least distance from the mean is said to be the matching distribution. All of the distributions are then sorted by a measure of ω_k/σ_k, and the weights are updated, increasing the matched distribution's weight and decreasing all others by factor of α, the learning rate.

$$\widehat{\omega}_{k,t} = (1 - \alpha_t)\omega_{k,t} + \alpha_t P(k|\mathbf{X}_t, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{19}$$

where $P(k|\mathbf{X}_t, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = 1$ if K is a match, else it is 0. The mean and standard deviation of the matched distribution are then updated as well:

$$\widehat{\boldsymbol{\mu}}_{k,t} = \big(1 - \rho_{k,t}\big)\boldsymbol{\mu}_{k,t} + \rho_{k,t}\mathbf{X}_t, \tag{20}$$

$$\widehat{\sigma}_{k,t}^2 = \big(1 - \rho_{k,t}\big)\sigma_{k,t}^2 + \rho_{k,t}\Big(\big(\mathbf{X}_t - \widehat{\boldsymbol{\mu}}_{k,t}\big)^{\mathrm{T}}\big(\mathbf{X}_t - \widehat{\boldsymbol{\mu}}_{k,t}\big)\Big), \tag{21}$$

$$\rho_{k,t} = \alpha \mathrm{N}\big(\mathbf{X}_t|k, \boldsymbol{\mu}_{k,t}, \boldsymbol{\Sigma}_{k,t}\big). \tag{22}$$

If the matched distribution's weight is above a set threshold, T, it is part of the background; else the pixel is classified as foreground. In the case of no match, the pixel is classified as foreground the worst fitting distribution is replaced with initial values for ω_k and σ_k, with the $\boldsymbol{\mu}_k$ parameter set to the current pixel value. Derivation of the update equations has not been given by [27], [77], but the algorithm can be interpreted as a K-means approximation of the result achieved by EM for GMMs using causal low pass filtered adaptations for GMM parameters. This process has been summarized graphically in Figure 4.

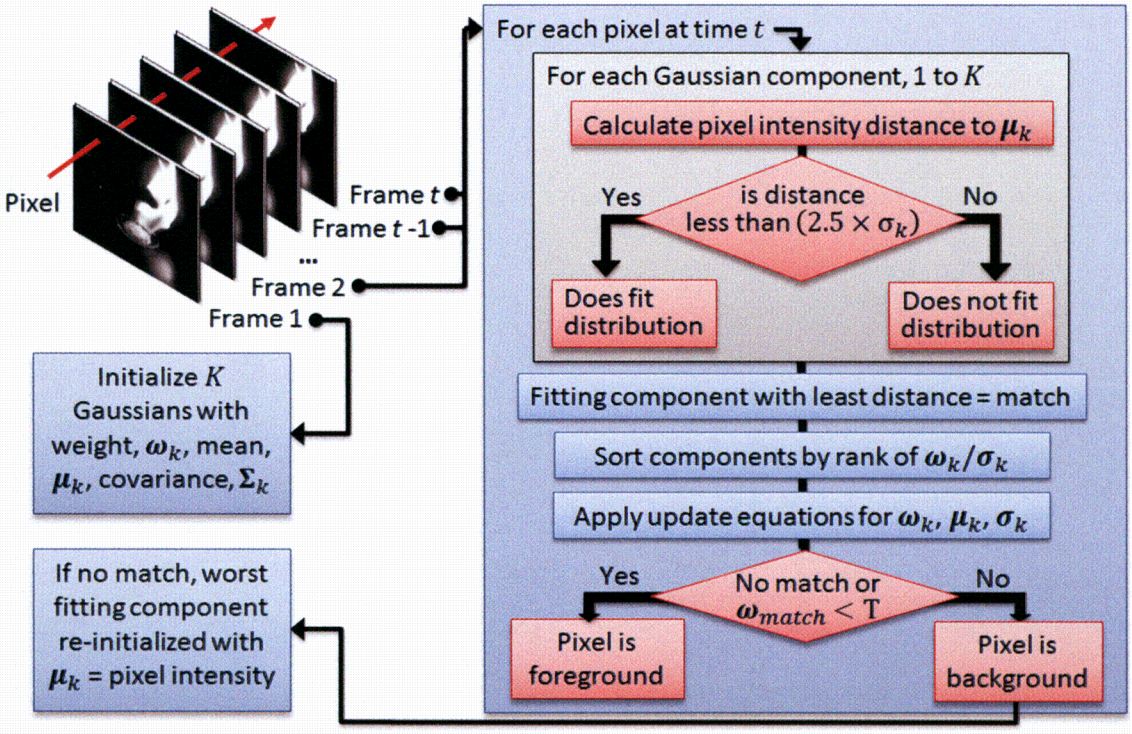

Figure 4. Adaptive GMM foreground segmentation algorithm flow chart.

2.4.3.2 Predictive Filtering

A signal processing approach to estimating the background of an image sequence involves the use of a Kalman filter. A Kalman filter is capable of estimating the state of a system

based on previous measurements. The details of the Kalman filter will be discussed in detail in section 2.6.5 as it has been applied to the task of object tracking. In the case of background estimation, the measurements used for updating the estimate for each new image in the sequence are the intensities of each pixel at the current time step. Using this estimate, the background can be modeled and adapted at each time step to changing background conditions. The works presented by Ridder, Munkelt, and Kirchner [79], Karmann and Brandt [80], and Boninsegna and Bozzoli [81] are successful applications of Kalman filter based foreground segmentation.

An application of Wiener filtering is used in the method of background modeling developed by Toyama, Krumm, Brumitt, and Meyers [82]. Their Wallflower algorithm models each pixel in an image sequence independently of one another using a one-step Wiener filtering process. The Wiener filter predicts the value of the next pixel based on a retained window of previous pixel measurements. This is similar to the Kalman filtering method, except that the discrete Kalman filter does not require a window of past measurements to estimate the future state of the system.

2.4.3.3 Nonparametric Kernel Density Estimation

A generalized method of modeling the distribution of any type of data is through the use of nonparametric kernel density estimation. This method works in the case of background modeling by providing a way to learn the distribution of each pixel in a sequence of images. Based on this background model, reasonable change in the value of that pixel intensity can be determined to segment the foreground. Kernel density estimation, also known as Parzen Windows density estimation, uses a kernel function (usually a Gaussian

function) to weight the distribution at each point of available data. This weighting value is determined by the combined distances from each of the surrounding data points in the series. Using this method, the correct distribution of any set of data can be determined but it relies intrinsically on the availability of an extremely large number of data samples. This becomes a problem when dealing with a real-time video analysis system, because it is not feasible to maintain all of the past data in a sequence of images. To combat this problem, as employed by Elgammal et al. [83], a weighting function gives greater contribution to recently seen images than that of past images.

Another problem with this method is computational complexity since density estimation must be completed for each pixel over a retained sufficient number of samples at each image in the sequence. Also, since only a finite history of samples can be used at each pixel, the bandwidth of the kernel ("window width") must be optimized in order to give the best approximation of the distribution. This will always be an approximation since the sequence of images cannot be predicted. A general value must provide a tradeoff between bias and variance of the estimated distribution from the true distribution.

2.4.3.4 Neural Network Modeling

Another method of developing a background model of a scene is to employ the use of neural networks for learning the previously seen distributions of pixel values in the image sequence. The SOBS (Self-Organizing Background Subtraction) algorithm [84] works on the basis of using a neural network grid that classifies each pixel based on its hue, saturation, and value to determine whether the pixel of a newly updated frame is classified as the pixel in that location of the background of the scene. This works in parallel within each

pixel in the image. Parameters can be adjusted for how much change must be detected and how long it takes to merge the current scene with the background estimate. The results of the SOBS algorithm are quite comparable to the results of the Pfinder [75], VSAM [16], and Codebook [85] that are known to be robust and also diverse in their structures. The SOBS system does a better or equal job in detecting motion than all of the other algorithms, but at the cost of computational complexity. However, it was not so computationally intensive that it could not be used for real time video segmentation. The size of the video frames plays a large role in this complexity. Sequences were used that ranged from simple to complex in terms of lighting and shadows as well as camouflaged motion.

In the work of Culibrk et al. [86], a neural network is created in order to estimate the PDF of the pixel changes in intensity in an image. The neural network classifies the most recent pixels as foreground or background based on the previously seen statistics for a time dependent on the learning rate of the BNN (Background modeling neural network). It is a combination of a PNN and a winner take all neural network. The classifier is updated automatically over time to update the background estimation. The method works fairly well, especially considering the complexity of the algorithm. It is capable of performing its calculations in parallel, which makes it easier to integrate into embedded systems such as in camera hardware.

2.4.3.5 Median Filtering

As an alternative to temporal differencing, median filtering is a more robust technique that maintains low complexity compared to advanced statistical methods. The general

approach is to retain a buffer of previous images in the sequence and calculate the median of the intensity values for each pixel in the image. This median image becomes an approximation of the background for the retained previous images. An approximated median filtered background model can also be used in order to avoid keeping a buffer of images. This reduces memory usage as well as allowing all previous images to have an effect on the current median background image rather than only the buffer images. The approximation is updated simply by incrementing the median estimate at each pixel in the image by one if the current pixel is larger than the estimate, and decreasing the median estimate if the current pixel is smaller than the estimate. The estimate will ultimately be an approximation of the median for each pixel. The median image can then be used to determine if a change has occurred beyond a set threshold, representing a foreground pixel [87].

2.5 Morphological Processing

The result of foreground segmentation is a binary image with white pixels representing areas of motion in video and black pixels representing the background. Even for a video containing a simple background, the segmented foreground is usually not ideal. Mathematical morphological operations are performed in order to enhance the segmented objects and remove artifacts that are not useful for tracking and classification.

2.5.1 Connected Component Analysis

Before anything useful can be done with the foreground image, it must be analyzed to determine connected components, or foreground objects, that may exist. Connected

components in a binary digital image represent regions exhibiting connectivity as specified by 4-neighbor or 8-neighbor connectivity as shown in Figure 5. Connectivity is defined by how neighboring white (binary 1 valued) pixels are located in proximity to each other. In a 4-neighbor connectivity scheme, two white pixels are part of the same connected component if they are located either directly North, South, East, or West of each other on the image grid.

	$(i, j-1)$	
$(i-1, j)$	(i, j)	$(i+1, j)$
	$(i, j+1)$	

$(i-1, j-1)$	$(i, j-1)$	$(i+1, j-1)$
$(i-1, j)$	(i, j)	$(i+1, j)$
$(i-1, j+1)$	$(i, j+1)$	$(i+1, j+1)$

Figure 5. 4-neighbor connectivity (left) and 8-neighbor connectivity (right) of pixel at image index (*i, j*).

8-neighbor connectivity is defined similarly, except that two white pixels may be located North, Northeast, East, Southeast, South, Southwest, West, or Northwest of each other to be connected. Every pixel in a connected component need not be neighboring every other pixel in the connected component, but there must be a path connecting neighboring pixels to all other pixels in the connected component. Additionally, all 8-neighbor connected components are by definition 4-connected as well. Figure 6 displays an example of 4-neighbor connectivity and how it differs from 8-neighbor connectivity.

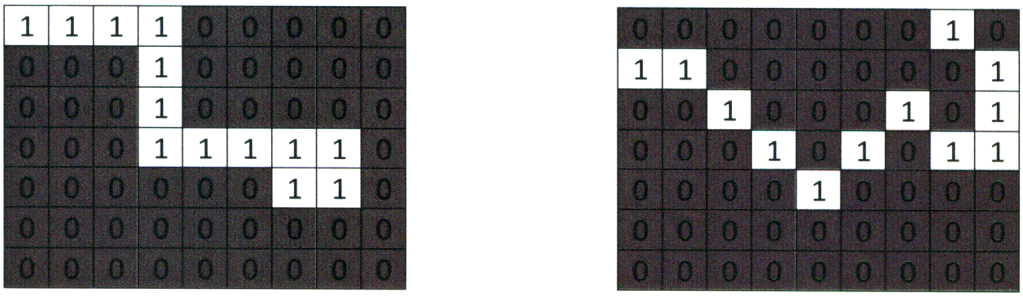

Figure 6. Example of 4-neighbor connected component (left) and 8-neighbor connected component (right).

2.5.2 Morphological Open

The morphological opening operation of processing on the foreground segmented image is a morphological erosion followed by a dilation. The morphological erosion and dilation operations require the use of what is known as a structuring element to determine the manner in which the operation is performed. The erosion operation effectively shrinks the borders of each connected component by removing the overlapping portions of the structuring element as its center is placed around each position of the perimeter of the connected component. In terms of mathematical set theory, this operation is defined as

$$A \ominus B = \{z | B_z \subseteq A\}, \tag{23}$$

the erosion of A by B is "the set of all points z such that B, translated by z, is contained in A", where A is the set of foreground white pixels and B is the set of structuring element pixels [88]. The dilation operation acts oppositely to erosion, expanding the borders of each connected component by adding the non-overlapping portions of the structuring element as its center is placed around each position of the perimeter of the connected component. Mathematically, dilation is defined as

$$A \oplus B = \{z | [B_z \cap A] \subseteq A\}, \tag{24}$$

"the set of all displacements, z, such that B and A overlap by at least one element," where A is the set of foreground white pixels and B is the set of structuring element pixels [88].

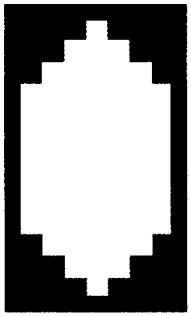

Figure 7. Structuring element used for morphological dilation and erosion.

The structuring element shape chosen for this application is a pointed rectangle similar to an ellipse with the major axis in the vertical orientation as seen in Figure 7. The reasoning for this choice is that flames and smoke tend to rise due to gases heated from combustion having less density than the surrounding air. Also, the primary nuisance source in a shipboard environment comes from personnel and their movements. The shape of a person is generally in the vertical direction, including the arms, legs, and head. This choice of structuring element accentuates vertical features, allowing for better foreground object recovery compared to a simple disc/circular element for example.

The ordered combination of the erosion and dilation is known as a morphological open, defined by

$$A \circ B = (A \ominus B) \oplus B, \tag{25}$$

where A is the set of foreground white pixels and B is the set of structuring element pixels [88]. It generally will smooth the contour of the connected component, remove lines or

isthmuses between connected components, and eliminate thin protrusions. Also during this step, small foreground objects and artifacts are removed that are considered too small to be classified. These artifacts can be considered noise, as they are false positives resulting from imperfections in image acquisition by the video source.

2.5.3 Morphological Close

A morphological dilation followed by erosion performed on an image is known as a morphological close. A close operation tends to smooth sections of connected component contours, combines areas with narrow breaks, removes small holes, and fills gaps in the contours. The morphological close operation can be defined alternatively as

$$A \cdot B = (A \oplus B) \ominus B, \tag{26}$$

where A is the set of foreground white pixels and B is the set of structuring element pixels [88].

2.6 Object Tracking

Persistently tracking foreground objects between frames is a crucial element of the work in this thesis. The calculation of features for tracked objects, particularly motion features, relies inherently on correctly matching a moving object to its previous position in a video stream. At each frame, after the foreground has been segmented and processed, each foreground object is labeled and its image centroid is calculated. Object labeling is performed based on connected component analysis. By matching objects between frames, motion-based features can be extracted from them.

2.6.1 Object Centroid

The calculation of the centroid of foreground objects is the first step required in tracking those objects between frames. The centroid of an image object is the center of mass of the object, i.e. the weighted average position based on all intensities in the image. Considering first a continuous 2D function, $f(x,y)$, the centroid is determined by combining central moments calculated over the function. The moments, m_{pq}, of order $(p+q)$ are calculated by

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\, dx\, dy \tag{27}$$

for $p, q = 0, 1, 2, \ldots$. Central moments are calculated by moving the coordinate system to the center of the continuous function. The central moments are calculated by

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y)\, dx\, dy, \tag{28}$$

where the x component of the centroid is calculated by

$$\bar{x} = \frac{m_{10}}{m_{00}}, \tag{29}$$

and the y component of the centroid is calculated by

$$\bar{y} = \frac{m_{01}}{m_{00}}. \tag{30}$$

In this case we are dealing with a discretely valued 2D function, a digital image, $f(x,y)$. The central moments of a digital image are determined by

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y). \tag{31}$$

Some of the higher order central moments will be used as extracted features for classification. These will be discussed in detail in section 2.7.3.1.

2.6.2 Distance Matching

In order to determine how an object is to be matched with itself in a previous frame, a distance measure must be used. Each object's centroid position is calculated from all other object centroids in the preceding frame. The object that has the least squared Euclidean distance (L2 Norm) metric is determined to be the match. Tracking parameters will be updated using the updated position. The L2 Norm is given by

$$\|\mathbf{x}_t - \mathbf{x}_{t-1}\| = (\mathbf{x}_t - \mathbf{x}_{t-1})^{\mathrm{T}}(\mathbf{x}_t - \mathbf{x}_{t-1}), \tag{32}$$

where x_t is the centroid position $\langle x, y \rangle$ at time, t.

2.6.3 Kalman Filtered Position

The object centroid position measurements used for distance calculations are approximations that are innately noisy; therefore there is great benefit in applying a filter to increase accuracy and reduce fluctuations caused by the motion segmentation process. A linear predictive Kalman filter/estimator is applied to the centroid position measurements for each foreground object. The Kalman filter provides a means for inferring missing information from these noisy measurements, effectively predicting the most likely future position based on previous states [89]. The state-space model for each foreground object is updated as a discrete dynamic system, assumed to be linear. The system is governed by the linear stochastic difference equation

$$x_k = Ax_{k-1} + Bu_k + w_k, \tag{33}$$

which attempts to estimate the state, x, of the dynamic process with a measurement, z, given by

$$z_k = Hx_k + v_k, \tag{34}$$

where w_k and v_k are the process and measurement noise assumed to be independent of each other, white, and of Gaussian probability distributions with covariance matrices Q and R respectively. "A in (33) relates the state at the previous time step, $k-1$, to the current time step, k. B in (33) relates the optional control input, u, to the state, x. H in (34) relates the state to the measurement, z_k." $A, B, H, Q,$ and R are assumed in this case to be constant over all k [90].

The discrete Kalman filter is recursive and involves a measurement update step (correction step) and a time update step (prediction step). The time update equations are given by (35) and (36), and the measurement update equations are given by (37), (38), and (39). The recursive Kalman filtering process is illustrated graphically in Figure 8.

1) Time Update

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1} \tag{35}$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \tag{36}$$

2) Measurement Update

$$K_k = P_{k|k-1}H^{\mathrm{T}}\left(HP_{k|k-1}H^{\mathrm{T}} + R\right)^{-1} \tag{37}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(z_k - H\hat{x}_{k|k-1}\right) \tag{38}$$

$$P_{k|k} = (I - K_kH)P_{k|k-1} \tag{39}$$

The measurement update is completed by first computing the Kalman gain given by (72). The next step is to obtain a measurement/observation (in the case of this thesis it

is the foreground object centroid position), z_k as modeled in (69), and then compute the *a posteriori* state estimate using (73). Finally, the *a posteriori* error covariance is calculated using (74). The process is then repeated using the previous *a posteriori* estimates to predict the new *a priori* estimates over each frame that the foreground object's position is tracked in a video stream [90].

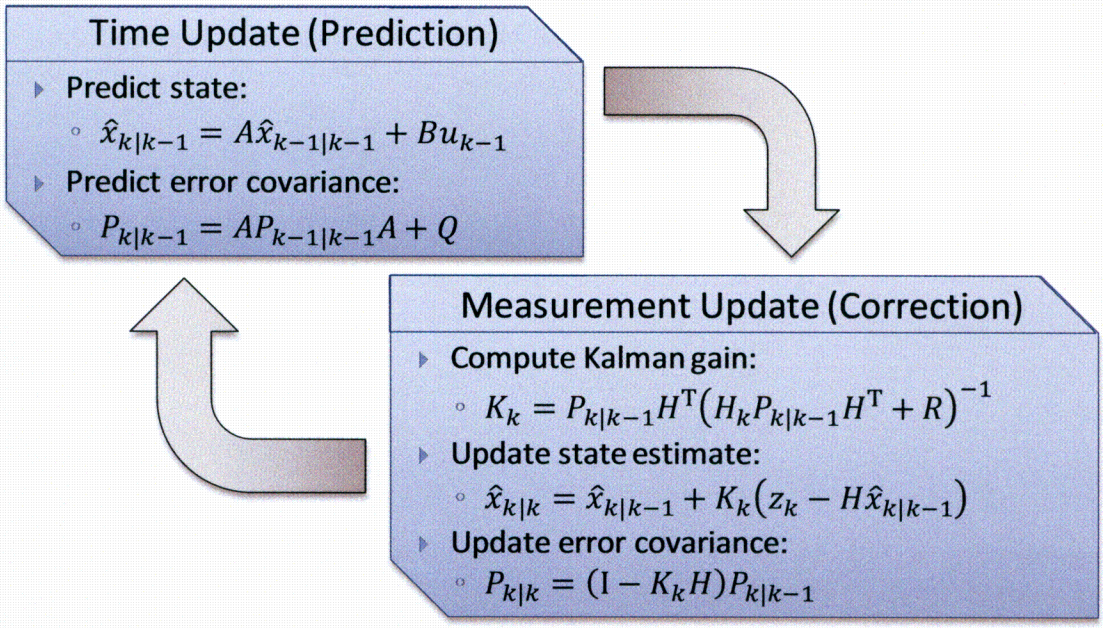

Figure 8. Discrete Kalman filter process.

In the application of foreground object position tracking, Kalman filtering is employed due to its ability to predict the position of a foreground object. Since tracking is dependent on matching the current position of a foreground object to the previous position, if the current position can be predicted then the capability to correctly match the object between frames is facilitated. The filtered state of the system is the true state of the object centroid's x and y position coordinates on the image plane with error

covariance of the filter storing the relationships between previous measurements and the previous prediction most representative of the true state of the system.

2.7 Object Feature Extraction

In the active area of video smoke and fire detection research, it is still unknown what types of features hold the most useful information for discriminating between nuisance and anomalous events. This section explains the many types of features that have been examined in this thesis. Each feature is calculated for foreground segmented objects that are successfully tracked for a sufficient number of consecutive frames. Features have also been normalized to frame height and width whenever possible in order to be as invariant to transformation as possible.

2.7.1 Shape-Based Features

There are several shape-based features/descriptors that can be calculated for foreground segmented binary objects. Although smoke and fire may not always assume the same shape, these features may be useful in capturing generalizations about their shape and distinguishing between that of nuisance objects.

2.7.1.1 Area

The area of each foreground object is calculated for each frame that is successfully tracked. The area of the binary image is equivalent to the central moment μ_{00} defined generally by (66). It is the total number of white pixels defined by the foreground object, $f(x, y)$, given specifically by

$$\mu_{00} = \sum_x \sum_y f(x,y), \tag{40}$$

where $f(x,y) = 1$ for foreground pixels and $f(x,y) = 0$ for background pixels. This area is then normalized to the total pixel area of the entire frame,

$$A_{obj} = \frac{\mu_{00}}{A_{frm}}, \tag{41}$$

where

$$A_{frm} = H_{frm} \times W_{frm}. \tag{42}$$

The normalized object area is a feature that is invariant to translation and rotation.

2.7.1.2 Perimeter

The perimeter is a boundary shape descriptor calculated for each foreground by determining the distance around the outer contour. This distance is found by summing the distances between the centers of all pixels lying on the boundary of the object as depicted in Figure 10.

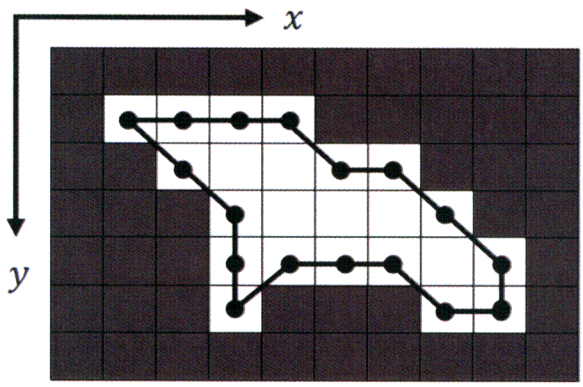

Figure 9. Perimeter around 8-connected binary foreground object.

If we define x_i and y_i to be the coordinates of the i^{th} pixel that forms the boundary of the object [91], then the perimeter is given by

$$P_{obj} = \sum_{i} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}. \quad (43)$$

2.7.1.3 Compactness

Combining the regional area descriptor with the perimeter boundary descriptor, a new dimensionless feature can be calculated known as compactness. Compactness provides a measure of the efficiency with which a boundary encloses an area [91]. It is calculated by

$$C_{obj} = \frac{P_{obj}^2}{\mu_{00}}. \quad (44)$$

2.7.1.4 Bounding Box Height and Width

The bounding box of a foreground object can be defined as the smallest rectangle that can contain the object. It is commonly used for drawing attention to tracked objects in surveillance videos for display purposes. The height and width of this rectangle are found by determining the pixels in the foreground object that are at the upper, lower, left, and right extremes of the object's contour. The offset between extremity pixel coordinates in the x and y directions can be used to determine the smallest rectangle that contains the object as seen in Figure 11. The height and width of this rectangle are then extracted as features used for object classification.

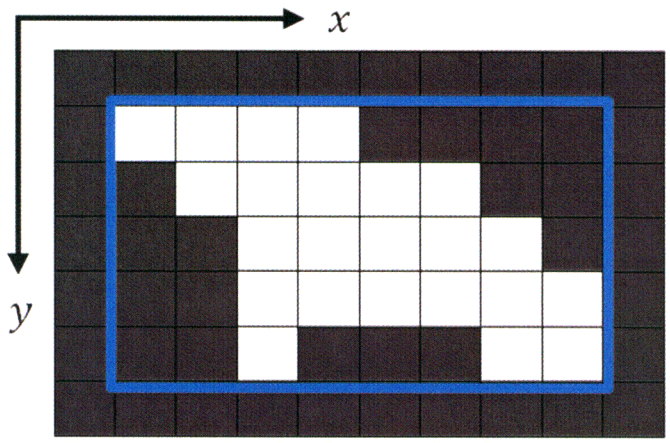

Figure 10. Bounding box of foreground object.

The extracted bounding box height is then normalized to the height of the video frame by

$$\widehat{H}_{box} = \frac{H_{box}}{H_{frm}}. \tag{45}$$

The extracted bounding box width is normalized to the width of the video frame by

$$\widehat{W}_{box} = \frac{W_{box}}{W_{frm}}. \tag{46}$$

2.7.1.5 Aspect Ratio

The bounding box height and width features may be further used for calculation of the aspect ratio descriptor. Aspect ratio is invariant to changes in both scale and translation. It is a dimensionless quantity defined as the ratio of the bounding box height to the bounding box width, given by

$$AR = \frac{H_{box}}{W_{box}}. \tag{47}$$

2.7.1.6 Extent

Using the area of the foreground object as well as the height and width of the bounding box containing the object, another feature known as extent can be calculated. It is defined as the ratio of the foreground object area to the total bounding box area given by

$$Extent = \frac{\mu_{00}}{A_{box}}, \tag{48}$$

where

$$A_{box} = H_{box} \times W_{box}. \tag{49}$$

2.7.1.7 Major and Minor Axis Length

The major and minor axes lengths of the ellipse having the same second order moments as the foreground object are also calculated. Using central moments defined by equation (66) in 2.6.4, normalized by the number of object pixels, the lengths are calculated. The moments are used directly in the equations defined for an ellipse, even though the shape is assumed not to be an ellipse. This measure is similar to the bounding box height and width, except that it is calculated based on the shape of the object and not the fixed image coordinate system. The central moments are normalized by

$$\eta_{xy} = \frac{\mu_{xy}}{N}, \tag{50}$$

Where N is the number of object pixels. The major axis length is calculated as in [92] by

$$Major\ Axis\ Length = 2\sqrt{2}\sqrt{\eta_{20} + \eta_{02} + \sqrt{(\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2}} \tag{51}$$

and minor axis length is calculated by

$$Minor\ Axis\ Length = 2\sqrt{2}\sqrt{\eta_{20} + \eta_{02} - \sqrt{(\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2}}. \quad (52)$$

2.7.1.8 Eccentricity

The extracted eccentricity feature is the ratio of the distance between the foci and major axis length of the ellipse with the same normalized second order central moments as the foreground object. It is a measure of how much a shape deviates from being perfectly circular. A circle has an eccentricity value of zero, and a line segment has an eccentricity value of one. This descriptor is calculated by

$$Eccentricity = \frac{2\sqrt{\left(\frac{Major\ Axis\ Length}{2}\right)^2 - \left(\frac{Minor\ Axis\ Length}{2}\right)^2}}{Major\ Axis\ Length}. \quad (53)$$

2.7.1.9 Orientation

The orientation descriptor is given by the angle between the x axis of the image coordinate system and the major axis of the ellipse with the same second order moments as the object. The orientation, θ, is calculated using normalized central moments as described in 2.7.1.7, and is given by

$$\theta = \begin{cases} \tan^{-1}\left[\frac{\eta_{02} - \eta_{20} + \sqrt{(\eta_{02} - \eta_{20})^2 + 4\eta_{11}^2}}{-2\eta_{11}}\right], & \eta_{02} > \eta_{20} \\ \tan^{-1}\left[\frac{-2\eta_{11}}{(\eta_{20} - \eta_{02}) + \sqrt{(\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2}}\right], & \eta_{02} \leq \eta_{20} \end{cases} \quad (54)$$

2.7.2 Spatiotemporal Features

The next class of object features that were extracted for classification were those fully utilizing the tracking capabilities of the algorithm. The tracking capabilities allow for time

varying features to be determined, capturing changes in region based features that may be considered useful for classification.

2.7.2.1 Growth

One particular feature that is of considerable interest for video fire and smoke detection is the growth of a foreground object. Growth is not difficult to calculate but may hold considerable discriminating information in detecting smoke and fire regions over nuisance objects. Smoke and fire will nearly always start very small at a single source and develop in the incipient stage to the smoldering and flaming stages as discussed earlier in section 2.2. The natural progression of a fire is a growth in size of smoke and/or fire regions over time. Human activity does not convey this property generally because although they are dynamic objects in video, they are static in terms of overall area seen by the camera except during possible occlusion of objects in the field of view. The growth feature, or change in normalized area, is given by

$$Growth(k) = \frac{A_{obj}(k) - A_{obj}(k-1)}{A_{obj}(k-1)}, \tag{55}$$

where k represents the index of the current image in the sequence.

2.7.2.2 X/Y Delta

The Kalman filtered centroid positions of all foreground objects are tracked at each frame, therefore the change in these positions can be determined as well. The change in position in the x and y directions are calculated by

$$\Delta xpos_{obj}(k) = xpos_{obj}(k) - xpos_{obj}(k-1), \tag{56}$$

$$\Delta ypos_{obj}(k) = ypos_{obj}(k) - ypos_{obj}(k-1). \tag{57}$$

where k represents the index of the current image in the sequence. These values are then normalized by frame dimensions using

$$\Delta\widehat{xpos}_{obj}(k) = \frac{\Delta xpos_{obj}(k)}{W_{frm}}, \tag{58}$$

$$\Delta\widehat{ypos}_{obj}(k) = \frac{\Delta ypos_{obj}(k)}{H_{frm}}. \tag{59}$$

2.7.2.3 X/Y Velocity

The velocity of each foreground object is calculated by using the change in Kalman filtered centroid position in the x and y directions and the known time interval between frames. This feature is thought to be able to capture the slow moving nature of flames and smoke in indoor environments such as a shipboard environment. It also may capture faster moving objects such as humans walking and interacting, which are nuisance events. The velocities of the foreground object are given by

$$v_x(k) = \frac{\Delta\widehat{xpos}_{obj}(k)}{\left(\frac{1}{FPS}\right)}, \tag{60}$$

$$v_y(k) = \frac{\Delta\widehat{ypos}_{obj}(k)}{\left(\frac{1}{FPS}\right)}, \tag{61}$$

where FPS is the known frames per second of the source video stream or image sequence.

2.7.3 Statistical Features

In addition to shape and spatiotemporal features, statistical features are calculated in order to further determine possible distinguishing foreground object descriptors. Although not specifically tailored to the detection of smoke and fire, they may hold

significance as they are quite universal in all types of applications, capable of providing a global description of an image region.

2.7.3.1 Hu's Invariant Moments

Continuing the discussion of statistical moments from section 2.6.4, higher order moments can provide descriptive capability for an image when combined in the proper manner. Central moments, described by (66), allow statistical moments to be invariant to translation by transforming the image coordinate system to a local coordinate system determined by the location of the image centroid. In order to achieve greater invariance, the central moments must be further subjected to normalization by the image area, μ_{00}. This concept and the set of seven higher order moments invariant to translation, rotation, and changes in scale were derived by Hu. Hu's invariant moments are calculated using combinations of second and third order normalized central moments [88]. The normalized central moments are given by

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \tag{62}$$

where

$$\gamma = \frac{p+q}{2} + 1 \tag{63}$$

for $(p+q) = 2, 3, \ldots$. Hu's seven invariant moments are then calculated by

$$\phi_1 = \eta_{20} + \eta_{02}, \tag{64}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2, \tag{65}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2, \tag{66}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2, \tag{67}$$

$$\begin{aligned}\phi_5 = {} & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2],\end{aligned} \tag{68}$$

$$\begin{aligned}\phi_6 = {} & (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ & + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}),\end{aligned} \tag{69}$$

$$\begin{aligned}\phi_7 = {} & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].\end{aligned} \tag{70}$$

The seventh invariant moment, ϕ_7, is also skew invariant [88], [91], [92]. These seven invariant moments were calculated not only for each binary foreground object, but also on the gray level image contained within the foreground object in the most recent frame. This gray level image represents the source frame with the binary object applied as a mask over the frame, ignoring all pixels except in the region of the foreground object.

2.7.3.2 Gray Level Statistics

By using the gray level image masked from the current frame by a binary foreground object as described in the previous section, the gray level pixel intensities can be used for general statistical calculations. These calculations may be thought of as texture statistics that capture information about the distribution of intensities in the region. The features for gray level histogram statistics are also based on calculations of moments. The difference from the other moment calculations is that they are now one dimensional, describing a set of intensity values. The n^{th} moment of a random variable, z, denoting gray levels is given by

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i), \tag{71}$$

where $p(z_i)$, for $i = 0, 1, 2, \ldots, L-1$, is the gray level histogram of z, L is the total number of unique possible gray levels, and m is given by

$$m = \sum_{i=0}^{L-1} z_i p(z_i), \tag{72}$$

representing the mean value of z, or average gray level of the region. The second moment is a measure of the variance of gray levels contained in the object mask, describing image contrast. The third moment is a measure of the skewness of the gray level histogram. This will reveal whether or not the distribution is weighted more heavily on the left or right side of average gray level (dark or light in appearance). A perfectly symmetrical histogram would have a skewness value of zero. Although not used in this thesis, the fourth moment can be related to a measure of the flatness of the histogram [88].

The variance, or second order moment, of the gray level intensities can also be used to define a measure of smoothness [88] given by

$$S = 1 - \frac{1}{1 + \hat{\mu}_2(z)}, \tag{73}$$

where

$$\hat{\mu}_2(z) = \frac{\mu_2(z)}{(L-1)^2}, \tag{74}$$

the variance normalized between 0 and 1. Another histogram feature extracted is known as uniformity, given by

$$U = \sum_{i=0}^{L-1} p^2(z_i). \tag{75}$$

The average entropy of the image region can also be calculated as a feature by

$$e = -\sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i). \tag{76}$$

2.7.4 Spectral Features

The spatial frequency components of the foreground segmented objects have also been considered as features for classification. 2D Discrete Cosine Transform (DCT) coefficients have been extracted from both binary foreground objects and their corresponding masked gray level regions. The DCT is a unitary transformation that has had primary application in image compression due to its capability for energy compaction. The 2D DCT of a discrete image, $f(i,j)$, of size $N \times N$ is calculated by

$$\mathcal{F}(x,y) = C(x)C(y) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) \cos\left[\frac{(2j+1)y\pi}{2N}\right] \cos\left[\frac{(2i+1)x\pi}{2N}\right], \tag{77}$$

where

$$C(k) = \begin{cases} \sqrt{\frac{1}{N}} \text{ for } k = 0 \\ \sqrt{\frac{2}{N}} \text{ for } k = 1, 2, \ldots, N-1 \end{cases}. \tag{78}$$

The output of the transform is a set of coefficients that provide a compact representation of the image in the DCT domain. The coefficients can be thought of as the numerical weights that when multiplied by the standard DCT basis patterns, return the transformed

image [93], [88], [94], [95]. The DCT basis patterns for an 8x8 image are shown in Figure 13 [94]. The patterns are combinations of horizontal and vertical cosine functions in the spatial domain, with increasing frequency as the row and column index increases. The first 10 DCT coefficients are retained for each binary foreground object and respective masked gray level region, given by the "zig-zag" zonal mask shown in Figure 12. The zonal mask retrieves the DCT coefficients in increasing order of spatial frequency, moving from low frequency components outward toward high frequency components [95]. This is believed to capture the low frequency components of smoke regions. Smoke blurs out detail in a scene, removing much of the spectral energy from the high frequency regions of the DCT image.

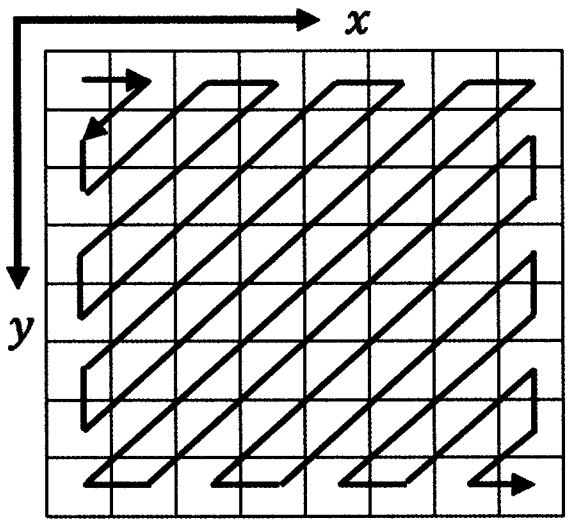

Figure 11. Zig-zag scan pattern.

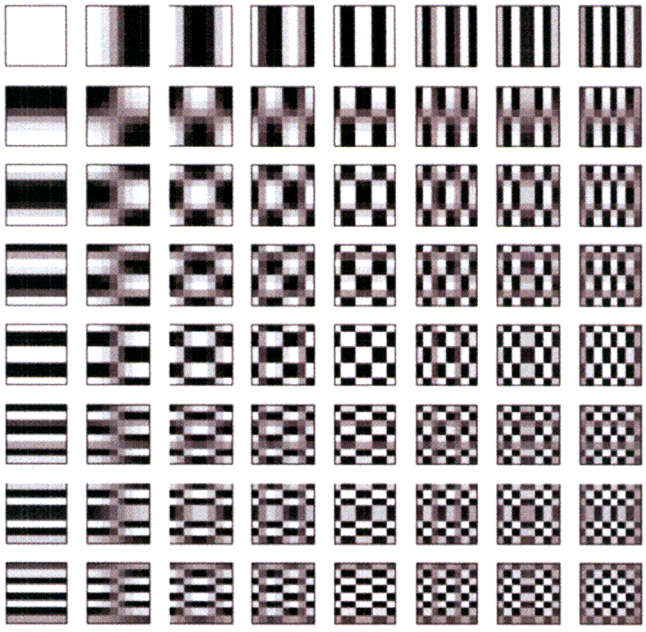

Figure 12. 8x8 DCT basis patterns.

2.8 Object Classification

In order for the algorithm to have some way of determining anomalous events from nuisance events, it must have some way of discriminating between observed features representing these events. The features extracted from foreground segmented and tracked objects are known to be either anomalous or nuisance. What is desired is the capability to classify the extracted features of all foreground objects generally. Methods of pattern recognition or machine learning can be used to train an algorithm with available training data. Training data in this case is the set of all extracted features of foreground objects, known to be of a particular class, either anomalous or nuisance.

2.8.1 Principal Component Analysis

The method of data pre-processing known as Principal Component Analysis (PCA) (also known as the Karhunen-Loéve transform) is a procedure that transforms the data in a way

that retains the most possible variance, while reducing the dimensionality of the data. The data in this case is the set of extracted feature vectors determined from the foreground segmented and tracked objects. The multidimensional feature vectors are transformed so that they are aligned with the principal axes of the data. The principal axes are the eigenvectors of the covariance matrix of that data with the largest corresponding eigenvalues [96].

In order to calculate the principal components, the vector representing the d-dimensional mean of the data set, $\boldsymbol{\mu}$, and the $d \times d$ covariance matrix, $\boldsymbol{\Sigma}$, are calculated for all data in the set. The eigenvectors, $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_d$, and eigenvalues, $\lambda_1, \lambda_2, \ldots, \lambda_d$, are calculated for the covariance matrix and sorted in descending order by eigenvalue. The k largest eigenvectors are chosen such that the dimensionality of the data will be reduced from d to k. A $k \times k$ matrix, $\mathbf{A}$, is then formed with the k eigenvectors as the columns. Prior to classification, the extracted feature vectors are then pre-processed by

$$\hat{\mathbf{x}} = \mathbf{A}^{\mathrm{T}}(\mathbf{x} - \boldsymbol{\mu}), \tag{79}$$

where $\mathbf{x}$ is a single d-dimensional feature vector [97].

2.8.2 Artificial Neural Network Classifier

The type of classifier used in this work is an artificial neural network (NN) classifier, and in particular the type is a multilayer perceptron (MLP) architecture. The NN classifier is an advanced pattern recognition algorithm that has been modeled by the way that the human brain functions. It is a general algorithm that has widespread application and has been referred to as a universal approximator, due to its capability to approximate any function of arbitrary complexity [98].

An MLP consists of d input nodes, h hidden layer nodes, and c output nodes. The MLP maps a d-dimensional input feature vector to one of c classes in a way that is determined by undergoing an offline training procedure. The training procedure is what determines the way that the input nodes, hidden layer nodes, and output nodes are interconnected. Based on this trained interconnectivity of nodes, the MLP may then classify newly presented feature vectors using the trained architecture. Depending on the availability of data and on the application, this classification may work remarkably well. Mathematically, the nodes of the network are connected via weighted, non-linear activation functions. The way that the weights are determined during the training procedure is by minimizing the gradient descent error of the criterion function, representing the difference between the desired output of the network and actual network output. The error is determined first at the output layer nodes, and is propagated backward to the hidden layer nodes. This type of training is known as back propagation, and has been utilized in this work [96].

CHAPTER 3: APPROACH

This thesis focuses on the design, development and validation of algorithms for the detection and tracking of anomalous events that can be identified from the analysis of monochromatic stationary ship surveillance video streams. The specific anomalies that have been focused upon are the presence and growth of smoke and fire events inside the frames of video streams. The objectives of this research are to:

1. Compile a survey of existing techniques for analyzing shipboard video stream data;
2. Design and develop a video foreground segmentation algorithm for determining regions of interest in video streams;
3. Design and develop an object tracking system capable of persistently tracking objects between frames;
4. Identify distinct and robust features from the tracked objects for detection and classification of anomalous indications;
5. Exercise the algorithm on a database consisting of canonical and experimental videos streams embedded with known anomalies (smoke and fire) as well as benign content;

The approach taken for this application is one of a generalized surveillance analysis system, tailored specifically to the detection of smoke and fire in video streams. The algorithm consists of the following steps as outlined in Figure 13.

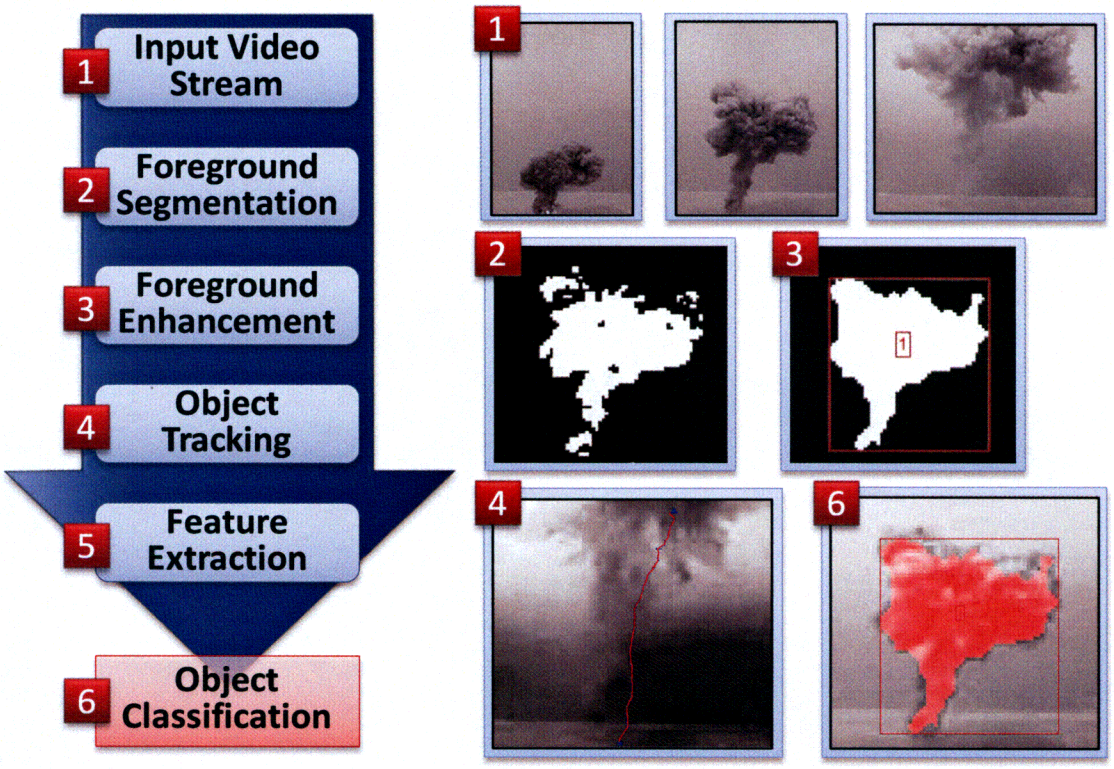

Figure 13. Approach for video analysis and anomalous event detection.

3.1 Foreground Segmentation

The second objective of this thesis is to design and develop a video foreground segmentation algorithm for determining regions of interest in video streams. Foreground segmentation of video is crucial for ensuring that only the regions of movement are analyzed and classified. In order to detect anomalous activity, the activity must be separated from the scene components that are not of interest. The approach taken for segmenting these foreground regions is to use a statistical background modeling algorithm to determine when motion has occurred. The background is modeled on a per-pixel basis using adaptive GMMs. Significant change in pixel intensity based on the learned intensity distribution determines whether or not a pixel is labeled as part of the

scene background or part of the moving foreground. Since the distribution is based on GMMs, complex multi-modal backgrounds can be learned by the algorithm. This method also allows the sensitivity of the foreground segmentation to be adjusted so that artifacts resulting from low resolution or compressed video streams can be minimized.

A modification to this algorithm has been applied for avoiding the problem of requiring long video lead time for learning the static scene background. The learning rate for all pixels during the first n frames of video is set to a large value so that within this short lead time, the background is learned very quickly. After n frames have been processed, the learning rate, α, is returned to the default initial value which will adapt the background model at a much slower rate, adjusting for illumination changes or a changing scene background.

3.2 Foreground Enhancement

The result of foreground segmentation is a binary image with white pixels representing foreground regions and black pixels representing the scene background. The foreground can be enhanced by applying morphological operations to the image. Connected component analysis is performed on each foreground segmented frame of video, using 8-pixel connectivity. The foreground objects determined by connected component analysis are first processed by using a morphological open operation, followed by a morphological close. The morphological opening and closing is achieved using a structuring element that accentuates rising smoke and upward reaching flames as well as the vertical shape of humans, the primary nuisance source studied in this work. Only foreground objects

containing a minimum number of pixels after this processing are retained. This eliminates small artifacts that may have resulted from the foreground segmentation process.

3.3 Object Tracking

In order to calculate spatiotemporal features for the segmented foreground objects, they must be tracked persistently between frames. The connected component analysis performed previously allows each object to be tracked independently. The position of the centroid of each object is stored between frames. This position is Kalman filtered in order to achieve the best possible estimate of the object's true position based on previous measurements. Objects are matched to their previous positions based on which object is the least far away in the previous frame based on spatial distance.

The tracking system provides the capability of calculating object growth features as well as object velocity. This has been exploited by allowing the velocity and growth of an object to influence the learning rate of the foreground segmentation algorithm, adjusted based on their values. Any foreground object that exhibits both slow moving and growing behavior will invoke a feedback loop to reduce the adaptive GMM algorithm learning rate for pixels contained within the region of the object. The learning rate, α, is maintained at the default value during the initial background learning process. Once a foreground object has been detected and tracked between successive frames, if the object velocity magnitude calculated by

$$\left|v_{obj}\right| = \sqrt{v_x^2 + v_y^2} \tag{80}$$

is below a specified minimum velocity, v_{min}, and the object growth calculated by (90) is greater than a specified minimum growth, $Growth_{min}$, then the feedback is initiated. Once initiated, the foreground segmentation learning rate parameter, α, for all pixels contained within the foreground object will be decreased by a specified step size, α_{step}, to a specified minimum learning rate, α_{min}. The learning at the object pixel locations will be increased by the same step size back to the initialized value if the object no longer exhibits slow movement or growth as specified previously, or if the object moves to a position no longer containing the affected pixel(s).

The adjustable learning rate is a solution to the problem of potentially anomalous foreground objects being absorbed into the background model of the scene too quickly. In the case of an incipient fire, it may take minutes for the anomaly to develop significantly enough for detection. Additionally, fire regions exhibit motion at some of the outermost portions but very little change is seen at the base and center of the fire. This feedback prolongs the detection of this type of anomalous region so that it may be tracked for a sufficient amount of time to extract features and classify the region.

3.4 Feature Extraction

After the foreground segmented objects have been tracked for a sufficient number of frames indicative of a significant event, features are calculated over each frame. Features are extracted based on shape, movement, gray level statistics, and spectral components. These features will be formed into a single vector and stored for classification purposes.

3.5 Object Classification

Finally, the extracted features corresponding to the anomaly are ranked based on their ability discriminate between anomalous and nuisance objects. This is done due to the uncertainty of which features are best suited for capturing discriminatory information for smoke and fire detection. A retained subset of all calculated features is subjected to Principal Component Analysis (PCA). The features extracted from all training data sequences are used for offline training of a MLP neural network classifier. The classifier is then used as the method of classifying newly presented video streams with features extracted for foreground objects, determining whether they are indicative of smoke and fire or are simply nuisance events.

CHAPTER 4: RESULTS

4.1 Implementation

The work performed in this thesis has been implemented in a manner that allows for a complete set of training videos to be analyzed and feature extracted for offline classifier training. The implementation treats each video as a sequence of images, simulating a streaming feed of newly acquired images by only allowing the algorithm to process one frame at a time. Figure 14 displays the overall block diagram for operation of all aspects of this thesis implementation. The implementation requires a few general inputs for proper operation. The height and width for which to resize the training video frames must be provided, which in the case of this implementation was 100px width and height scaled to maintain the same aspect ratio as the original source. The data directory where the training video data and configuration files are stored must also be entered. Each video must have a matching configuration file with a '.cfg' extension that provides all settings predetermined for that video. Most of the settings are related to the foreground segmentation algorithm to ensure that it will be optimal for the paired video. Additional parameters determine what types of results will be stored as well as whether robustness testing features will be applied to the source image. A list of all video file names that are to be analyzed is the final input that must be provided.

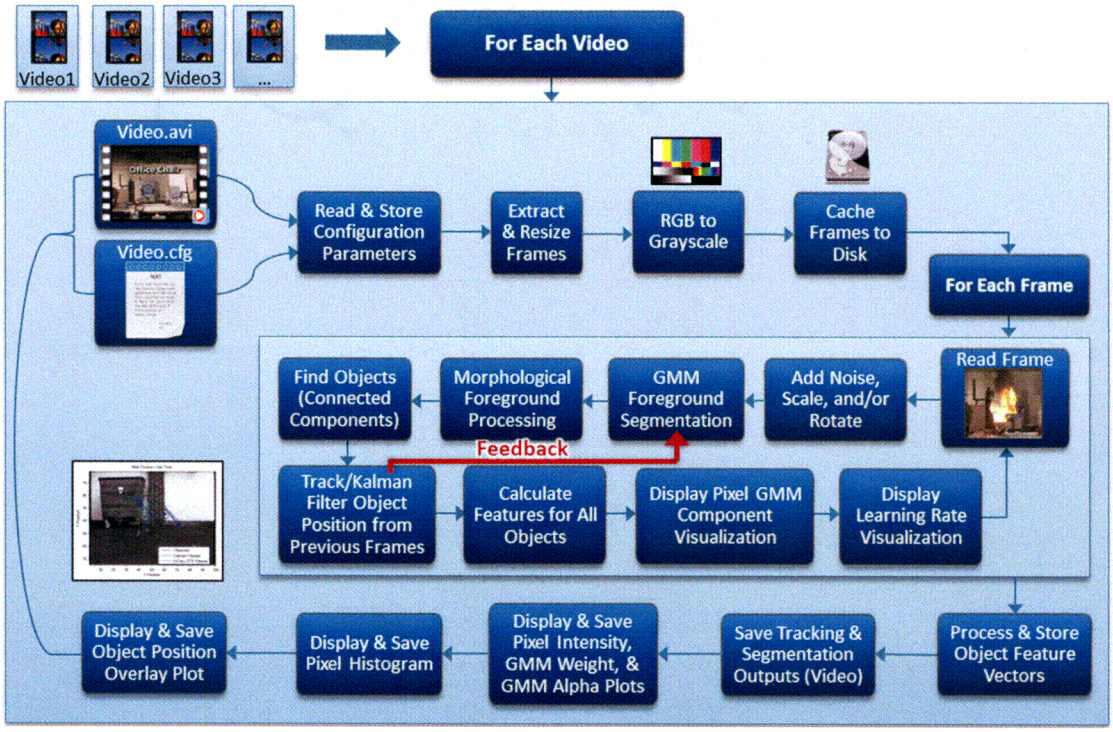

Figure 14. Surveillance video analysis automated feature extraction implementation block diagram.

At the start of the main analysis loop, the first block of processing will read the configuration file for the first specified video, storing all parameters into memory. The video file is then loaded and all frames within the specified range of total frames are extracted to a cache folder on the hard disk. Each frame is resized to the desired height and width and converted to gray scale. If the video has already been analyzed previously, the cache is used to speed up image retrieval.

The next analysis loop will cycle through all frames of the current video. The current frame is first loaded from the disk cache. If specified in the configuration file, robustness testing features can be applied to the frame before analysis. Gaussian, speckle, "salt & pepper" (impulse), or Poisson noise may be added to the frame. Horizontal or vertical sinusoidal interference in regular or random spatial frequencies may be added to

the frame. The frame can also be scaled or rotated inside the frame on a black background. Although testing was not performed using these functions, they are available for future use. Another possible use of these functions may be for bootstrapping the training data set to be tolerant of these perturbations and transformations. Figure 15 shows how these robustness testing features might be applied.

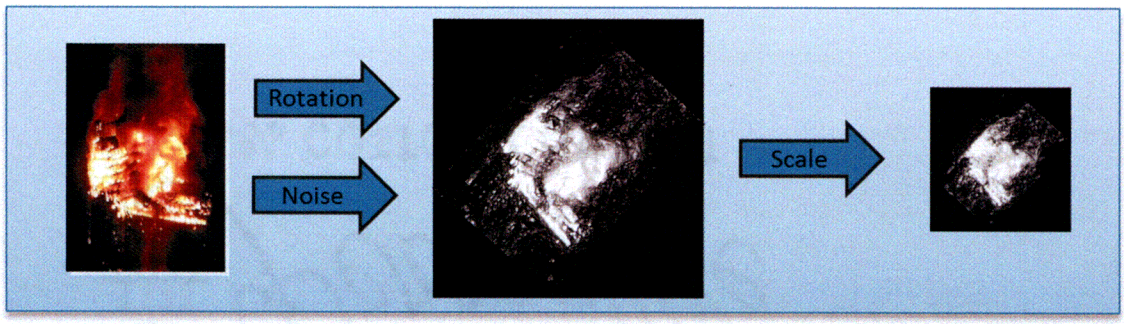

Figure 15. Example application of optional robustness testing features.

The most recent frame in the simulated video stream is then passed into the GMM foreground segmentation block, where a background model is initialized and updated at each pixel in the frame. Next, if the foreground segmentation output contains foreground pixels, morphological processing is performed. The foreground is enhanced and all connected components of size greater than the specified minimum number of pixels are labeled. These labeled foreground objects become the input to the next block of processing, which is the object tracking block. The tracking block maintains a history of the most recently seen objects, up to a maximum number specified in the configuration file. In this work, 5 is the maximum number of objects tracked between frames. The number of frames for which each object has successively been tracked is also maintained.

Upon first detection of a foreground object (or when the object cannot be matched to a previous frame), the initial position of the object is stored and all time-independent features are calculated for the object. Using the initial position, the Kalman filter model is initialized for the object. It should be noted that the optional Kalman filter control input model, B, and vector, u, were not used in this implementation. During successive detections of foreground objects, the tracking block attempts to match each current object to the object in the previous frame that has a centroid position with the minimum distance to the current object. If the closest object is less than a distance threshold specified in the configuration file, it will be matched, allowing for the Kalman filter model to be updated based on the new position, and time-dependent spatiotemporal features to be calculated for that object. The distance threshold is used to ensure that a foreground object will not be incorrectly matched to a newly appearing foreground object. The assumption is that foreground object motion is very small over each frame. As features are extracted for each tracked foreground object, they are stored in memory as a set of feature vectors specific to each object.

As foreground objects are tracked, their velocity magnitude and growth features are checked to determine whether or not they are indicative of a slow moving, growing object. The area of the object must increase by a specified percentage, $Growth_{min}$, from the previous frame (usually 5%), and also exhibit a velocity magnitude, $|v_{obj}|$ as calculated by (80), below a specified velocity threshold, v_{min} (usually $\sim 0.08 - 0.12$). This triggers the previously mentioned feedback loop to adjust the learning rate, α, of pixels

contained within the foreground object to decrease by a step size, α_{step} (usually $\sim 0.005 - 0.01$), toward a lower bound, α_{min} (usually ~ 0.01).

After all frames have been analyzed in the simulated video stream, all of the features extracted for the foreground objects that were tracked for a minimum number of successive frames are compiled into a single output and stored to disk. The minimum successive frame stipulation was implemented in order to ensure that the features extracted are for a significant event. The minimum number of successive frames used in this implementation was 20. Additionally, visualizations of foreground segmentation output, tracked objects, and pixel analysis are stored to disk after all frames are analyzed. The main analysis loop then continues to process all remaining videos specified as inputs to the system.

Classifier training is implemented as a separate offline process. A multilayer perceptron neural network classifier is trained using Levenberg-Marquardt backpropagation. The two layer network architecture contains one hidden layer with 5 hidden layer nodes. The activation transfer function used was the logistic sigmoid for the hidden layer nodes and the output layer nodes. Classifier performance was evaluated using K-fold cross validation, with 95% confidence intervals using critical t-score values for K-folds < 30 and critical z-score values for K-folds ≥ 30. Final classifier creation was done using 80% of the data set for training and 20% of the data set for tuning. The data set was also transformed using PCA to retain the 40 features (out of 58) with the most variance.

A final implementation was created as an optimized version of the feature extraction implementation. Any operation that was not necessary for feature extraction and classification was removed, such as the pixel analysis capabilities and additional visualizations. One added feature is the ability to trace out the desired region of the source video stream to be analyzed. Regions containing known nuisances can be ignored entirely such as windows or steam generators, etc. The trained neural network classifier is integrated into the final implementation and will classify foreground objects as they are detected. A pre-alarm condition is initiated upon first detection of smoke/fire. Upon 10 successive detections, the object is classified as smoke/fire. Otherwise it is classified as a nuisance object.

Both the feature extraction and final testing implementations have been created using the MATLAB software environment version R2009b with image processing and neural network toolboxes installed. An external Kalman filter toolbox [99] was used as well as the PRTools pattern recognition toolbox version 4.1 [100] for classifier training and analysis.

4.2 Training Database

A database of training videos to be used for the system feature extraction implementation was created as part of this thesis work. The database consists of 68 incipient fire/smoke and nuisance (human motion) video streams from various sources including NIST Building and Fire Research Laboratory (BFRL) fire tests, NAVSEA Philadelphia land based engineering site, CAVIAR [101] database, and ViSOR database [102]. A full list of training

videos and detailed information can be seen in Table 3, Table 4, Table 5, Table 6, Table 7, Table 8, Table 9, and Table 10.

#	Video Name	Description	Source	Training Type	Duration (h:m:s)	Width x Height	Frames	Training Range
1	Backlit_Smoke	Smoke machine in hallway with bright light in background	http://signal.ee.bilkent.edu.tr/VisiFire/	Smoke	00:00:28	320x240	483	1:185
2	Backlit_Smoke2	Smoke machine in hallway with bright light in background	http://signal.ee.bilkent.edu.tr/VisiFire/	Smoke	00:00:23	320x240	342	1:260
3	Barbeque	Flaming fire with little smoke outdoors in small grill	http://signal.ee.bilkent.edu.tr/VisiFire/	Fire	00:00:42	320x240	421	20:200
4	barium_chloride	Chemical flame test of barium chloride	YouTube	Fire	00:00:12	320x240	386	40:345
5	Black_Smoke_Masked	Green screened smoke plume rising upward	http://www.detonationfilms.com/Stock_Directory.html	Smoke	00:00:06	440x400	188	60:179
6	calcium_chloride	Chemical flame test of calcium chloride	YouTube	Fire	00:02:27	480x360	1923	1:1750
7	Camp_Fire	Flaming camp fire with dark background and slight camera motion	YouTube	Fire	00:00:26	480x360	674	1:640
8	College_Dorm_Outside	Outside of building where sprinklered college dorm test took place	NIST	Fire	00:03:33	420x280	3209	1:1100
9	College_Dorm_Smoke	Smoke accumulation in college dorm test	NIST	Smoke	00:02:07	290x280	1906	1:900
10	College_Dorm_Sprinklered	College dorm fire test extinguished with sprinkler	NIST	Fire	00:02:07	420x280	1906	1:250

Table 3. Video database items 1-10.

#	Video Name	Description	Source	Training Type	Duration (h:m:s)	Width x Height	Frames	Training Range
11	College_Dorm_Unsprinklered	College dorm fire test	NIST	Fire	00:01:23	420x280	1253	1:400
12	douglas_fir_1_5m	Douglas fir Christmas tree fire, 1.5m tall	NIST	Fire	00:00:58	240x420	884	1:340
13	douglas_fir_3_8m	Douglas fir Christmas tree fire, 3.8m tall	NIST	Fire	00:01:24	240x420	1272	1:329
14	douglas_fir_3m	Douglas fir Christmas tree fire, 3m tall	NIST	Fire	00:01:24	240x420	1272	1:350
15	Four_Workstations	Test conducted on office cubicles. Contains changes in illumination caused by smoke occlusion of light source	NIST	Fire	00:03:08	420x280	2834	130:250
16	GTG_particle_smoke	Gas Turbine Generator shaft in housing at NAVSEA LBES with CG particle smoke plume added	NAVSEA	Smoke	00:00:45	352x240	903	300:650
17	GTG_particle_dark_smoke	Gas Turbine Generator shaft in housing at NAVSEA LBES with dark CG particle smoke plume added	NAVSEA	Smoke	00:00:45	352x240	903	300:650
18	GTG_particle_white_smoke	Gas Turbine Generator shaft in housing at NAVSEA LBES with white CG particle smoke plume added	NAVSEA	Smoke	00:00:45	352x240	903	300:650

Table 4. Video database items 11-18.

#	Video Name	Description	Source	Training Type	Duration (h:m:s)	Width x Height	Frames	Training Range
19	MSA	Man walks into lobby, drops briefcase, walks back, stops, crouches, waves arms, and exits	http://cvprlab.unipparthenope.it/	Nuisance	00:00:35	320x240	528	20:380
20	NIST_Christmas_Tree	Flaming test of Christmas tree in living room	NIST	Fire	00:00:48	320x240	1457	1:283
21	NIST_Cubicle	Test of cubicle fire near to camera	NIST	Fire	00:04:12	720x480	7570	1:321
22	NIST_Living_Room	Slowly growing fire beginning at couch in a living room	NIST	Fire	00:03:07	720x480	5619	1:1500
23	NISTIR_7468_304 interior	Bunk bed fire seen from doorway near floor. Illumination changes caused by smoke accumulation near window light source.	NIST	Fire	00:02:48	320x240	5056	1:3000
24	NISTIR_7468_304 interior2	Smoke slowly entering living room caused by fire in adjacent bedroom.	NIST	Smoke	00:02:39	320x240	4778	1640:3000
25	Office_Chair_Flame	Slowly growing fire at standard office chair	NIST	Fire	00:20:03	240x320	18048	1:1500
26	Office_Chair_Smoke_Shadow	Shadow of office chair fire and smoke	NIST	Smoke	00:20:01	240x320	18024	1:1500
27	potassium_chloride	Chemical flame test of potassium chloride	YouTube	Fire	00:00:10	320x240	321	40:321
28	Rag_Flames	Burning rags with black background, shown from ignition	http://www.detonationfilms.com/Stock_Directory.html	Fire	00:00:32	720x480	982	1:982

Table 5. Video database items 19-28.

#	Video Name	Description	Source	Training Type	Duration (h:m:s)	Width x Height	Frames	Training Range
29	Single_Workstation	Workstation fire without cubicle walls	NIST	Fire	00:29:59	480x300	26989	1:2200
30	Sky_Fire	Close up of rapidly growing fire with sky background	http://www.detonationfilms.com/Stock_Directory.html	Fire	00:00:10	720x480	304	50:304
31	Sky_Fire2	Close up of large flickering fire with sky background	http://www.detonationfilms.com/Stock_Directory.html	Fire	00:00:05	720x480	179	20:179
32	Sky_Fire3	Close up of large flaming fire with sky background	http://www.detonationfilms.com/Stock_Directory.html	Fire	00:00:07	720x480	237	20:237
33	Sled_Chair_Flame	Fire test on office sled chair, showing flame portion of fire without shadows.	NIST	Fire	00:24:59	300x360	22490	1:800
34	Smoke_Machine	Smoke generated from smoke machine in front of green screen with shadow	http://www.hollywoodcamerawork.us/downloads.html	Smoke	00:00:12	480x404	300	40:300
35	Smoke_Manavgat	Smoke plume from great distance above forest on mountainside.	http://signal.ee.bilkent.edu.tr/VisiFire/	Smoke	00:03:33	320x240	3201	100:1400
36	Smoke_Plume	Rising thin smoke on black background	http://www.detonationfilms.com/Stock_Directory.html	Smoke	00:00:07	720x480	212	50:212
37	sodium_chloride	Chemical flame test of sodium chloride	YouTube	Fire	00:00:11	320x240	354	50:320
38	strontium_chloride	Chemical flame test of strontium chloride	YouTube	Fire	00:00:08	320x240	240	1:240

Table 6. Video database items 29-38.

#	Video Name	Description	Source	Training Type	Duration (h:m:s)	Width x Height	Frames	Training Range
39	Structural_Separation1	Fire test between adjacent exterior building walls.	NIST	Fire	00:09:20	260x400	8409	1:1300
40	Structural_Separation2	Fire test between adjacent exterior building walls with large reflections	NIST	Fire	00:02:31	260x400	2275	1:1250
41	Tree	Large Christmas tree burn with flying debris	NIST	Fire	00:01:12	420x430	1086	1:250
42	Windy_Smoke	Smoke grenade in alley with swirling wind source	http://signal.ee.bilkent.edu.tr/VisiFire/	Smoke	00:01:31	320x240	916	1:850
43	wood_frame_structure_test-left	Smoke and fire coming from within structure, focusing on roof and horizon	NIST	Fire	00:15:14	480x320	13716	1:2450
44	wood_frame_structure_test-right	Smoke and fire coming from within structure, focusing on roof and horizon	NIST	Fire	00:15:14	480x320	13716	1:1600
45	intelligentroom_raw	Man walking around conference room	http://cvrr.ucsd.edu/aton/shadow/index.html	Nuisance	00:00:30	320x240	300	1:300
46	Laboratory_raw	Two people walking and running back and forth past laboratory cabinets	http://cvrr.ucsd.edu/aton/shadow/index.html	Nuisance	00:01:28	320x240	887	1:887
47	Meet_Crowd	Group of people walking through lobby	http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/	Nuisance	00:00:20	384x288	519	1:519

Table 7. Video database items 39-47.

#	Video Name	Description	Source	Training Type	Duration (h:m:s)	Width x Height	Frames	Training Range
48	OneLeaveShopReenter1cor	People walking through mall corridor	http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/	Nuisance	00:00:15	360x270	388	1:388
49	Walk3	People walking through lobby	http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/	Nuisance	00:00:56	384x288	1420	1:1420
50	Cam3	Person walking in and out of room with camera at counter height	http://www.openvisor.org/video_details.asp?idvideo=129	Nuisance	00:01:01	320x240	926	1:926
51	Cam4	Person walking and out of room with camera mounted high	http://www.openvisor.org/video_details.asp?idvideo=130	Nuisance	00:01:04	320x240	965	1:965
52	domotica_031007_07	Person walking in room and crouching behind box	http://www.openvisor.org/video_details.asp?idvideo=123	Nuisance	00:00:34	360x270	864	1:864
53	LeftBag	Person walking through lobby and dropping off and picking up bag	http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/	Nuisance	00:00:32	384x288	806	1:806
54	LeftBag_Pickedup	Person walking through lobby and dropping off and picking up bag	http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/	Nuisance	00:00:54	384x288	1233	1:1233
55	tower1_set2	College campus activity with people walking, cars driving by	http://www.cs.ucf.edu/~arslan/surveillance/index.htm	Nuisance	00:07:11	320x240	12919	1:12919
56	TwoLeaveShop2cor	Two people walk out of store at mall	http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/	Nuisance	00:00:24	292x220	600	1:600

Table 8. Video database items 48-56.

#	Video Name	Description	Source	Training Type	Duration (h:m:s)	Width x Height	Frames	Training Range
57	TwoLeaveShop2front	Two people walk out of store at mall	http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/	Nuisance	00:00:21	320x240	539	1:539
58	visor_1212733798908_camminata3	Person walking across view, stopping once	http://www.openvisor.org/video_details.asp?idvideo=246	Nuisance	00:00:11	320x256	278	1:278
59	visor_1197283980290_10_hangar	Smoke plume rising in airplane hangar	http://www.openvisor.org/video_details.asp?idvideo=173	Smoke	00:01:32	384x288	2315	1:2315
60	GTG16_nuisance	3 people interacting beside GTG1 housing, near to camera and only partially in field of view	NAVSEA	Nuisance	00:01:06	352x240	1320	110:1150
61	GTG25_nuisance	Person moving around beside GTG2, cleaning housing	NAVSEA	Nuisance	00:26:39	352x240	31980	2900:7201
62	GTG15_nuisance	Person working between GTG1 and GTG2, walking in and out of field of view from distance	NAVSEA	Nuisance	00:04:13	352x240	5060	1:5060
63	GTG15_nuisance 2	People walking around busy area of LBES behind GTG1	NAVSEA	Nuisance	00:11:14	352x240	13480	1:13480
64	GTG15_nuisance 3	Long sequence of people working around GTG1 near camera	NAVSEA	Nuisance	00:15:28	352x240	18575	1:18575
65	GTG36_nuisance	People walking in hallway between GTG3 and support columns	NAVSEA	Nuisance	00:00:44	352x240	880	1:880

Table 9. Video database items 57-65.

#	Video Name	Description	Source	Training Type	Duration (h:m:s)	Width x Height	Frames	Training Range
66	GTG36_nuisance 2	People walking in hallway between GTG3 and support columns	NAVSEA	Nuisance	00:00:33	352x240	660	1:660
67	GTG36_nuisance 3	People walking in hallway between GTG3 and support columns	NAVSEA	Nuisance	00:01:21	352x240	1620	1:1620
68	New_vs_Old_Roo m_Fire_Final	Comparison of old and modern furniture in an outdoor fire test	Underwriters Laboratory	Nuisance	00:05:54	320x240	10620	60:10620

Table 10. Video database items 66-68.

4.3 Foreground Segmentation

The work was first tested for the foreground segmentation algorithm. This became the basis for work that followed, allowing for analysis of moving foreground objects. In order to be certain that the design and development of a video foreground segmentation algorithm for determining regions of interest in video streams was satisfied, testing had to be performed. Results of testing were analyzed based on initially simple sequences to determine satisfactory performance.

An initial sequence that was analyzed consisted of a moving white square on a black background. Each frame in the sequence had a height and width of 100px, and the moving square's sides are 10px long. The black background pixels had a gray level of 0 and the white square pixels had a gray level of 255 using an 8-bit quantization scheme with $L = 256$ unique gray levels. The path of the moving square begins in the upper left corner of the image, moving down in the negative y direction by 10 pixel increments until it is no longer visible. The square then reappears at the top of the image, moved in the positive x direction by 10 pixels. The square then moves in the negative y direction once again. This process is repeated until the square has moved far enough in the positive x direction to where it is no longer visible. This sequence is shown graphically in Figure 16.

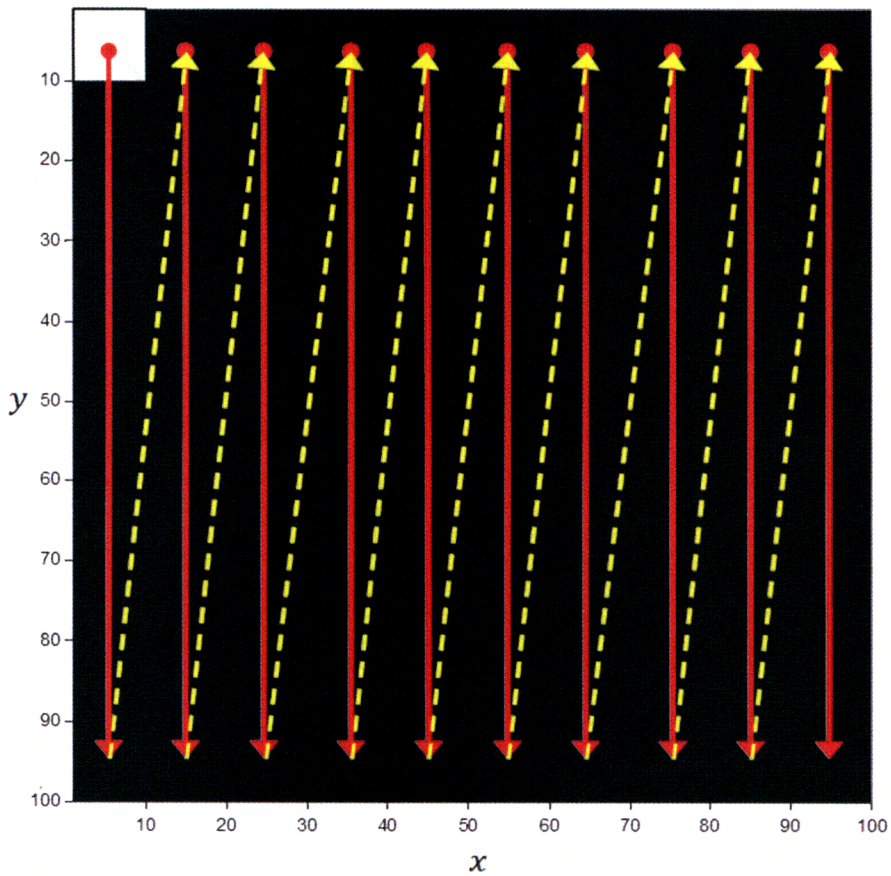

Figure 16. Foreground segmentation test sequence object movement path.

The parameters used by the foreground segmentation algorithm for these tests are displayed in Table 11.

Parameter	Description	Value
K	Number of Gaussians	3
T	Background weight threshold	0.4
α	Initial learning rate	0.1
μ_k	Initial mean	0
ω_k	Initial weight	0.05
σ_k	Standard deviation	9
σ_{max}	Standard deviation threshold	2.5

Table 11. Foreground segmentation algorithm parameters for test sequences.

The method of determining how well the foreground segmentation performs will be

based on the calculation of 3 different accuracy metrics independently at each frame [84]. Recall, also known as the detection rate, given by

$$Recall = \frac{TP}{TP + FN}, \tag{81}$$

where TP is the number of True Positive pixels. True positives represent the number of correctly detected foreground pixels, common to both the foreground segmentation output and the ground truth. The ground truth is the true measurement of the foreground, determined in this case at the time of sequence generation, but in some cases may be determined by a human segmenting the sequence by hand. FN is the number of False Negative pixels. False negatives represent the number of pixels in the ground truth not detected by foreground segmentation output. Precision, also known as positive prediction, given by

$$Precision = \frac{TP}{TP + FP}, \tag{82}$$

where FP is the number of False Positive pixels. False positives represent the number of pixels in the foreground segmentation output that are not common to the ground truth. The F1 score, or F-measure, is a weighted harmonic mean of precision and recall given by

$$F_1 = \frac{2 * Recall * Precision}{Recall + Precision}. \tag{83}$$

Each of the tested sequences reports the minimum, maximum, and mean value for recall, precision, and F_1 as calculated over each frame of the sequence. The results for the first sequence are shown in Table 12. The output of the foreground segmentation algorithm worked perfectly for the simple binary image sequence. The algorithm was then

tested using a slightly more complicated set of sequences. The same moving square animation was used as shown previously, but now the black background is replaced with a gray scale image of a fishing boat (Figure 17) taken from the USC SIPI image database, resized to 100px height and width. The intensity of the pixels in the square was also varied in each new sequence. A total of 18 new sequences were tested with a different gray level for the moving square in each sequence. The gray levels used are shown in Figure 18.

Sequence	Precision			Recall			F_1		
	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
White square against black BG	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 12. Foreground segmentation accuracy metrics for moving square sequence.

Figure 17. USC SIPI fishing boat image as gray scale background in test sequences.

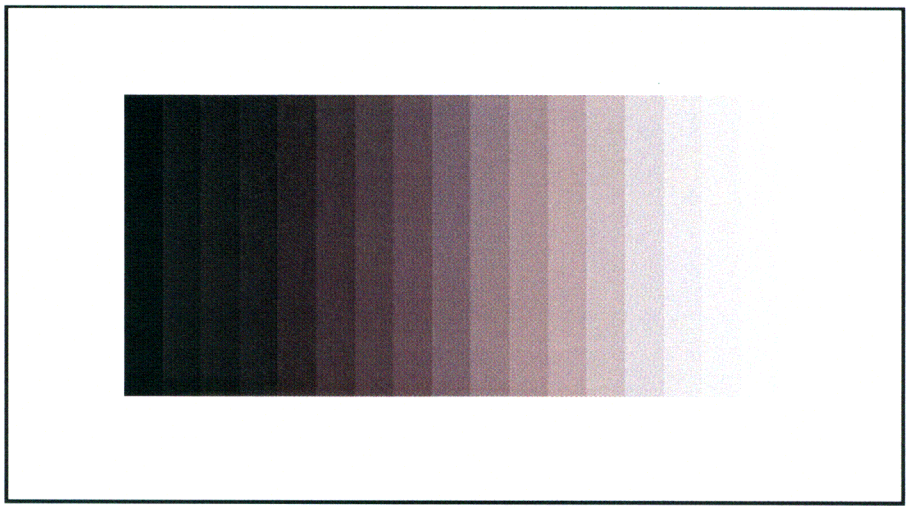

Figure 18. Gray levels of moving square in test sequences.

Sequence	Precision			Recall			F_1		
	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
Gray level 0 square	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Gray level 15 square	1.0	1.0	1.0	0.97	0.999	1.0	0.98477	0.99949	1.0
Gray level 30 square	1.0	1.0	1.0	0.95	0.9975	1.0	0.98477	0.99874	1.0
Gray level 45 square	1.0	1.0	1.0	0.95	0.9975	1.0	0.97436	0.99873	1.0
Gray level 60 square	1.0	1.0	1.0	0.97	0.9985	1.0	0.98477	0.99924	1.0
Gray level 75 square	1.0	1.0	1.0	0.98	0.9984	1.0	0.9899	0.99919	1.0
Gray level 90 square	1.0	1.0	1.0	0.96	0.9975	1.0	0.97959	0.99874	1.0
Gray level 105 square	1.0	1.0	1.0	0.96	0.997	1.0	0.97959	0.99848	1.0
Gray level 120 square	1.0	1.0	1.0	0.96	0.9942	1.0	0.97959	0.99707	1.0
Gray level 135 square	1.0	1.0	1.0	0.78	0.9861	1.0	0.8764	0.99281	1.0
Gray level 150 square	1.0	1.0	1.0	0.89	0.9798	1.0	0.9418	0.98967	1.0
Gray level 165 square	1.0	1.0	1.0	0.93	0.9911	1.0	0.96373	0.99548	1.0
Gray level 180 square	1.0	1.0	1.0	0.95	0.9973	1.0	0.97436	0.99864	1.0
Gray level 195 square	1.0	1.0	1.0	0.97	0.9991	1.0	0.98477	0.99955	1.0
Gray level 210 square	1.0	1.0	1.0	0.98	0.9986	1.0	0.9899	0.99929	1.0
Gray level 225 square	1.0	1.0	1.0	0.99	0.9997	1.0	0.99497	0.99985	1.0
Gray level 240 square	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Gray level 255 square	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 13. Foreground segmentation accuracy metrics for additional test sequences with fishing boat gray level background.

The results acquired using the more complicated test sequences were also overall very good. These tests turned out to produce examples where precision was perfect, but recall was not always perfect due to the presence of false negatives in the foreground segmentation output. Figure 19 shows exactly what causes these errors. As the square

moves across the image, there are regions of the image that have gray levels very similar to that of the square. This creates a situation where the change in gray level at certain pixels is not significant enough to be seen as a change from the previous frame so they are classified as background. In order for false positives to have been generated, some sort of morphological processing of the foreground image would have to create additional foreground area around the square.

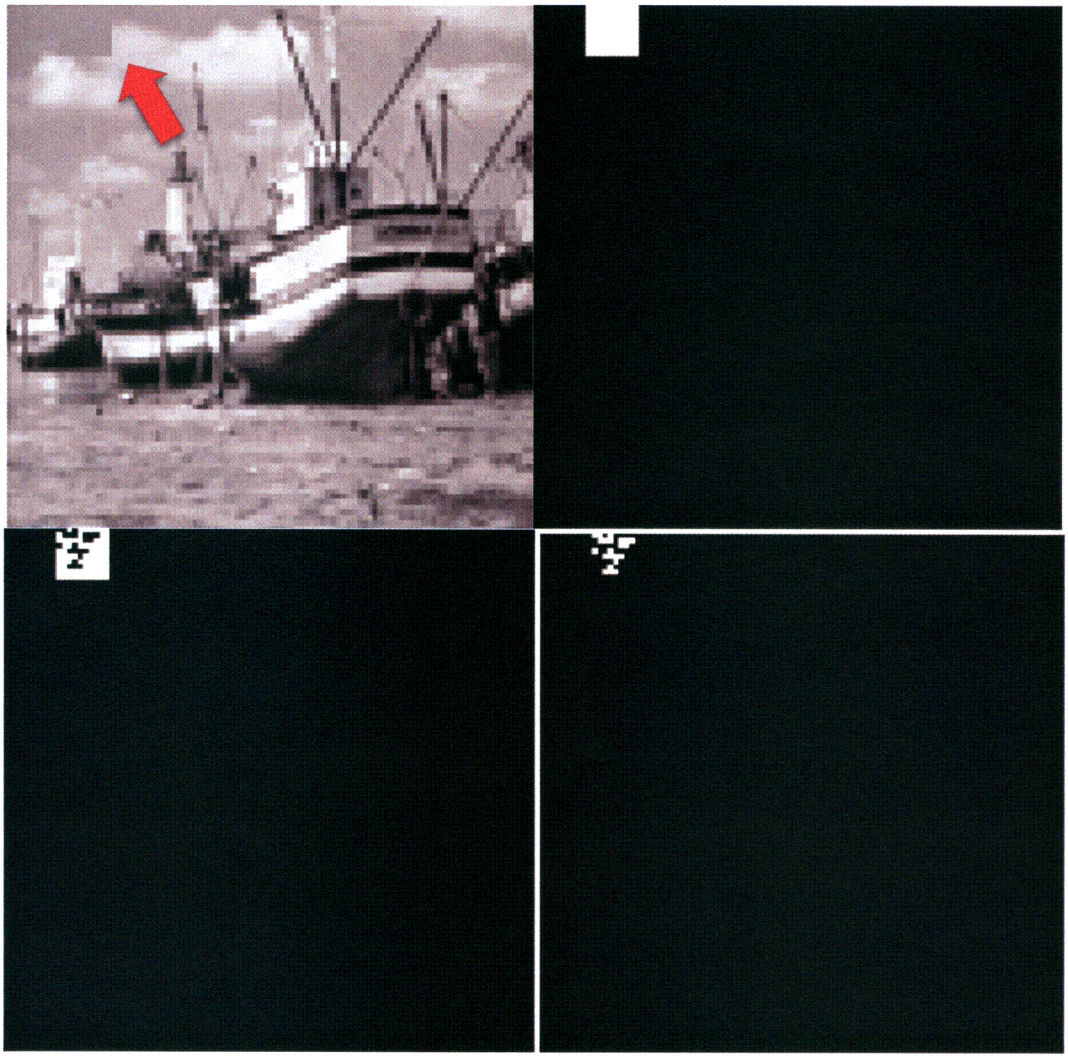

Figure 19. Moving square gray level 135 sequence frame 41 (top left), ground truth (top right), foreground segmentation output (bottom left), and segmentation false negatives (bottom right).

In order to minimize errors in the foreground segmentation output, it is necessary to fully understand the parameters used by the algorithm. Ultimately, repeated trial and error testing using real videos provided the best method of determining the correct parameters to be used. These parameters also may need to be adjusted slightly depending on the source video being analyzed. If the imagery contains low contrast, the sensitivity may need to be increased by reducing the standard deviation at each pixel GMM. If the background in the scene is complex, containing movement from something like moving water or trees, then the threshold for motion tolerance must be adjusted. A very dynamic scene with background objects changing state or illumination changes will need an increased learning rate so that these changes adapt quickly. The trade off with increasing the learning rate is then a reduced amount of time that a slow moving or stopped foreground object can remain in the foreground before being learned by the background model. As discussed previously, the learning rate problem has been addressed in this thesis by the tracking system providing feedback based on object velocity and growth feature calculation.

In an effort to speed up the estimation of parameters for videos being analyzed using the foreground segmentation algorithm, certain analysis capabilities were added to the implementation. The capability to visualize the state of the GMM distributions at any pixel (specified in configuration file) over the entire video has been implemented, as well as graphical representations of the pixel intensities (time-series and histogram), GMM weights, as well as adaptive learning rate. An example of the GMM distribution visualization can be seen in Figure 20, where the sequence analyzed was a flickering flame

from a camp fire with a black background. The visualization provides information for the pixel that has been surrounded by a blue box, shown in the original frame (top left), as well as the foreground segmented frame (after morphological processing, top right). Shown on the plot below these images is a visualization of the GMM components. In this case, $K = 3$, so 3 Gaussian components make up the GMM at each pixel as seen by the blue, red, and green bell curves.

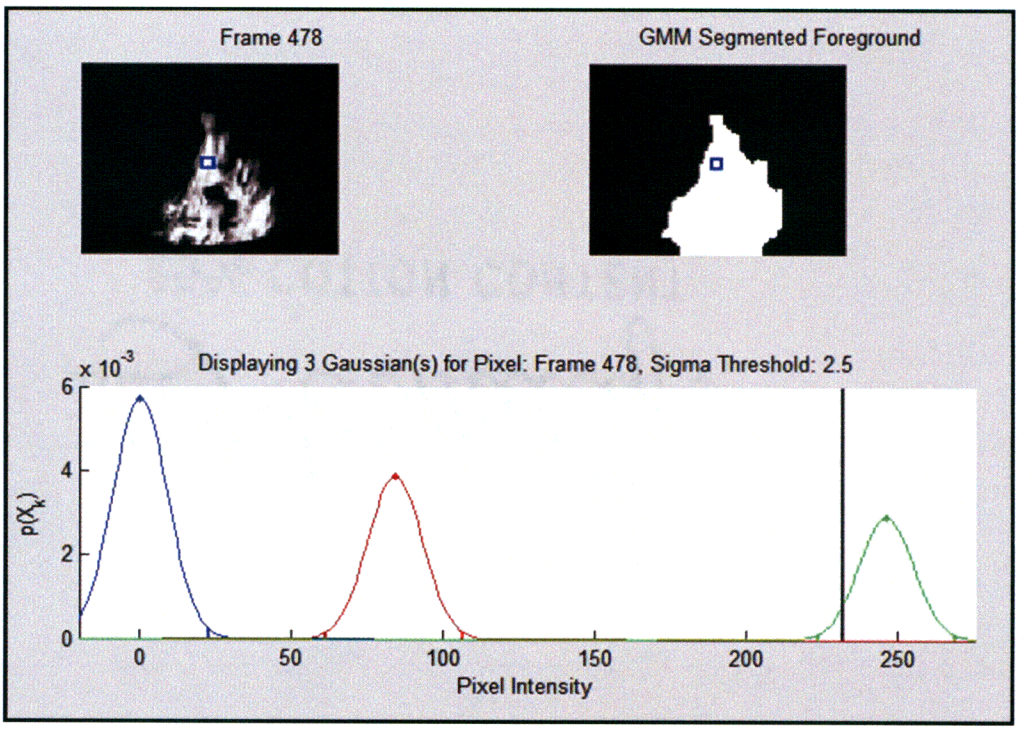

Figure 20. Pixel analysis GMM visualization for "Camp_Fire" sequence.

The blue curve is tallest, representing that it has the highest component weight, centered at gray level/intensity 0, the scene background. Since the pixel intensity is very dynamic due to being located at the edge of the flickering flame, it also has exhibited intensities centered around gray levels 80 and 250, indicated by the dot at the peak of the curves. The vertical black bar represents the current pixel intensity for the indicated

frame, and the smaller vertical bars under the tails of the curves represent the standard deviation threshold multiplied by the standard deviation, or the distance from the mean that a pixel's intensity must fall within to be considered a match for that distribution.

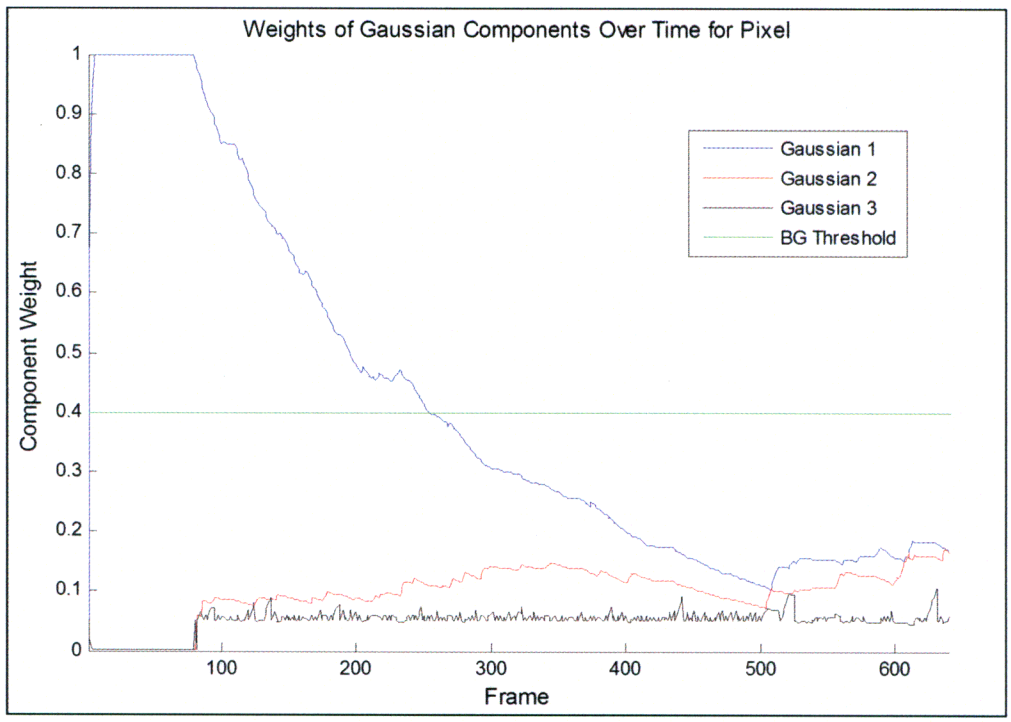

Figure 21. Pixel analysis GMM weight visualization for "Camp_Fire" sequence.

The weights of the Gaussian components at a single pixel can also be analyzed as in Figure 21 to determine if the threshold parameter (T) should be adjusted. In this case, the true gray level 0 background distribution (blue line) has the largest weight until around frame 510, where it meets the red line. The second most weighted distribution then becomes the new largest weighted distribution, switching back again where they cross near frame 610. This has occurred because the foreground object (flickering flame) has been slowly "absorbed," or learned, by the background model. This would pose a problem, except that the background threshold (green line) is set to a threshold above the

weights of each learned distribution. If none of the distribution weights are above the threshold (as in frames 275 or greater), then that pixel is in a state of uncertainty and will always be classified as a foreground pixel until one of the Gaussian component weights has increased to a value greater than the threshold and the current pixel intensity is matched to that distribution. Determining the threshold value to use will depend on the nature of the background within the video stream. Background activity can generally be known based on usual activity. An outdoor sequence with moving trees or ocean waves for example would contain a known dynamic background and as such would require calibration for parameter optimization.

The analysis of pixel intensity can also be done by looking at the time series graph of intensity over each frame as in Figure 22, and also the histogram of pixel intensities over all frames as in Figure 23. The flickering nature of the fire at the analyzed pixel is quite apparent due to the rapidly fluctuating intensity value. It can be seen in the histogram that the most frequently measured intensity is at the 0 gray level, which has been scaled down to show detail in the other histogram bins. Most of the other intensities are in the 200 to 250 gray level region. This visualization can reveal whether a background pixel is multimodal, which will determine the number of Gaussians that are optimal for the video.

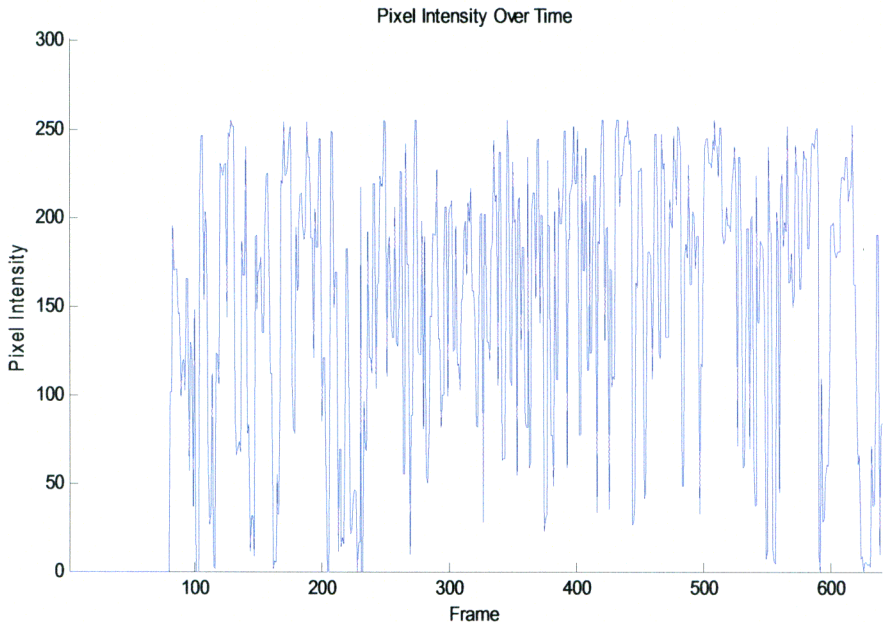

Figure 22. Pixel analysis intensity visualization for "Camp_Fire" sequence.

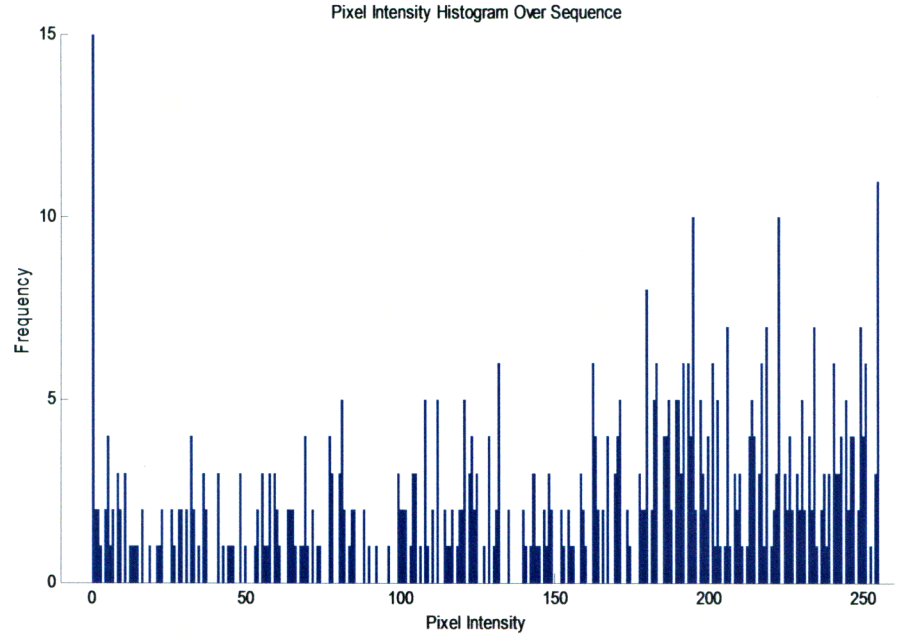

Figure 23. Pixel analysis intensity histogram for "Camp_Fire" sequence. Bin 0 has been scaled down to increase detail in other bins.

Since the background of this video is black, a single Gaussian may have been sufficient. Figure 24, Figure 25, and Figure 26 show additional training sequences with pixel analysis capabilities demonstrated.

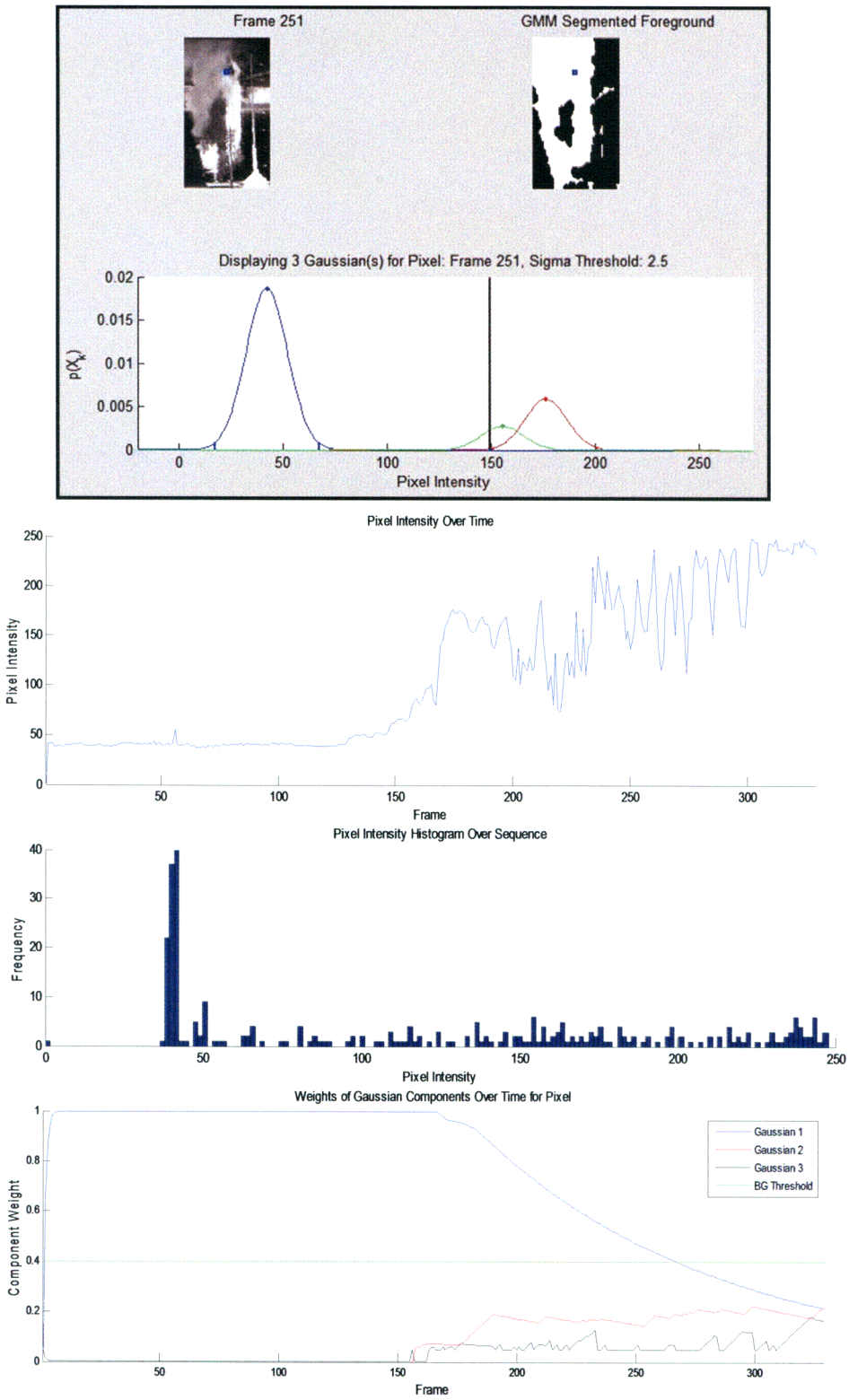

Figure 24. Pixel analysis for "douglas_fir_3_8m" sequence.

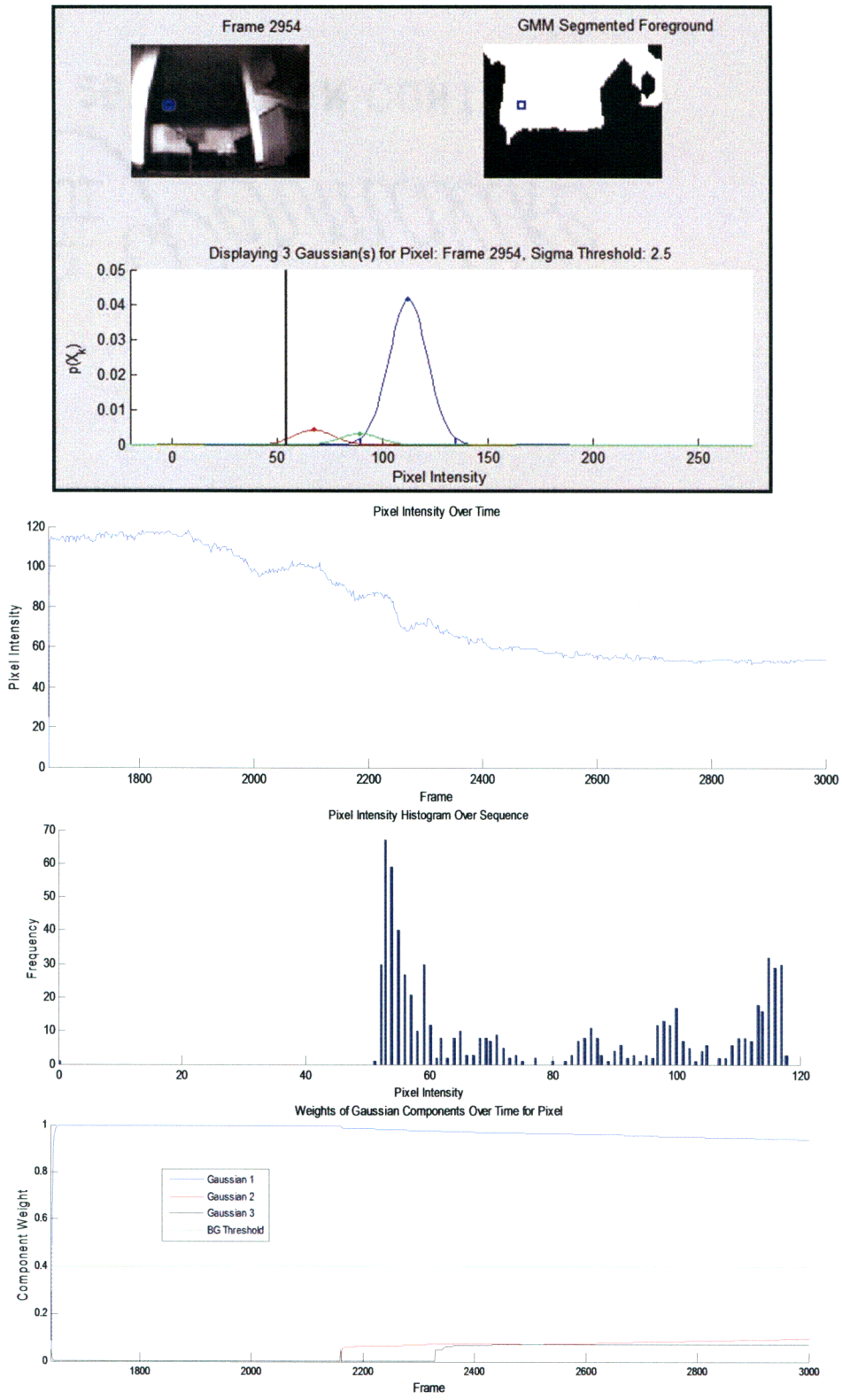

Figure 25. Pixel analysis for "NISTIR_7468_304interior2" sequence.

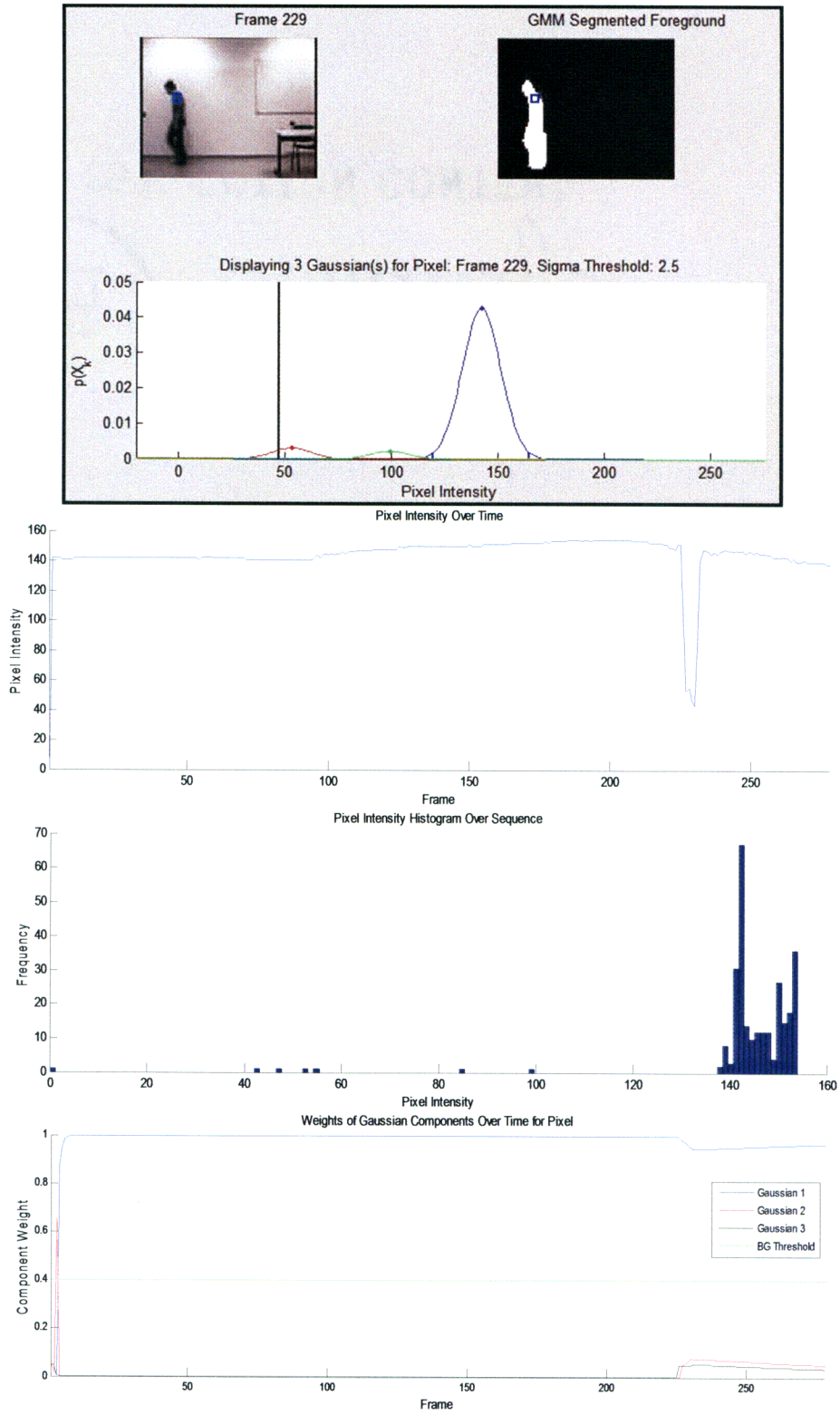

Figure 26. Pixel analysis for "visor_1212733798908_camminata3" sequence.

4.4 Foreground Enhancement

The output of the foreground segmentation algorithm is not going to be ideal for most videos. Morphological operations are performed to enhance the segmented foreground as much as possible. Figure 27 shows and example of how the foreground segmentation output may not be ideal. There are some small artifacts at the top of the foreground segmentation output, and many discontinuous regions throughout the main foreground object. During foreground enhancement, the small objects have been removed by performing connected component analysis. Also, morphological processing will smooth out the contours of the foreground as well as connect discontinuous regions. The enhanced foreground as a result is much more representative of the region of interest in the frame. Refer to Figure 28, Figure 29, Figure 30, Figure 31, Figure 32, and Figure 33 for further examples of foreground enhancement applied to training videos.

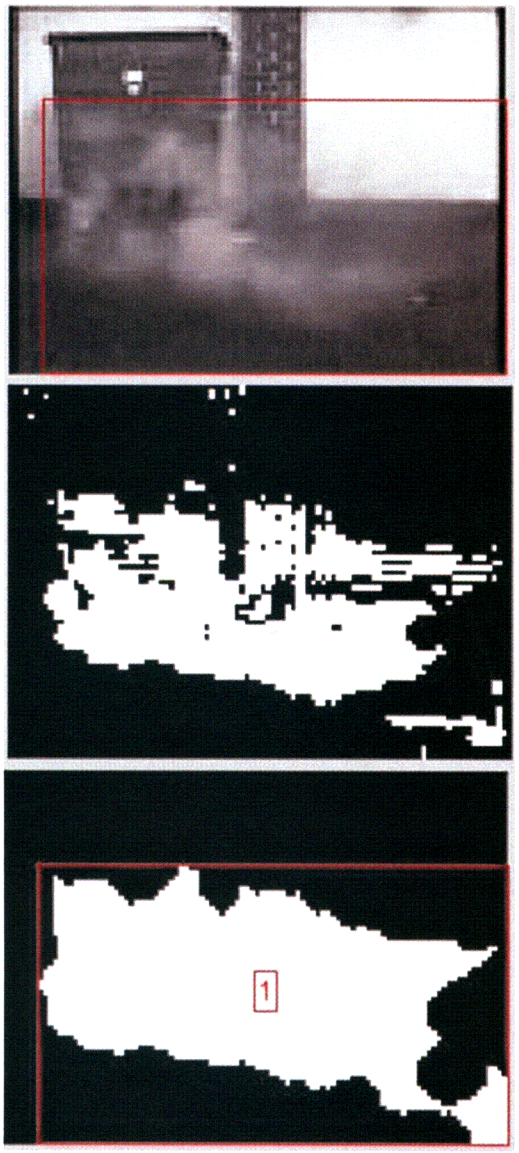

Figure 27. Bounded foreground object (top), foreground segmentation output (middle), and processed and enhanced foreground for "Windy_Smoke" sequence.

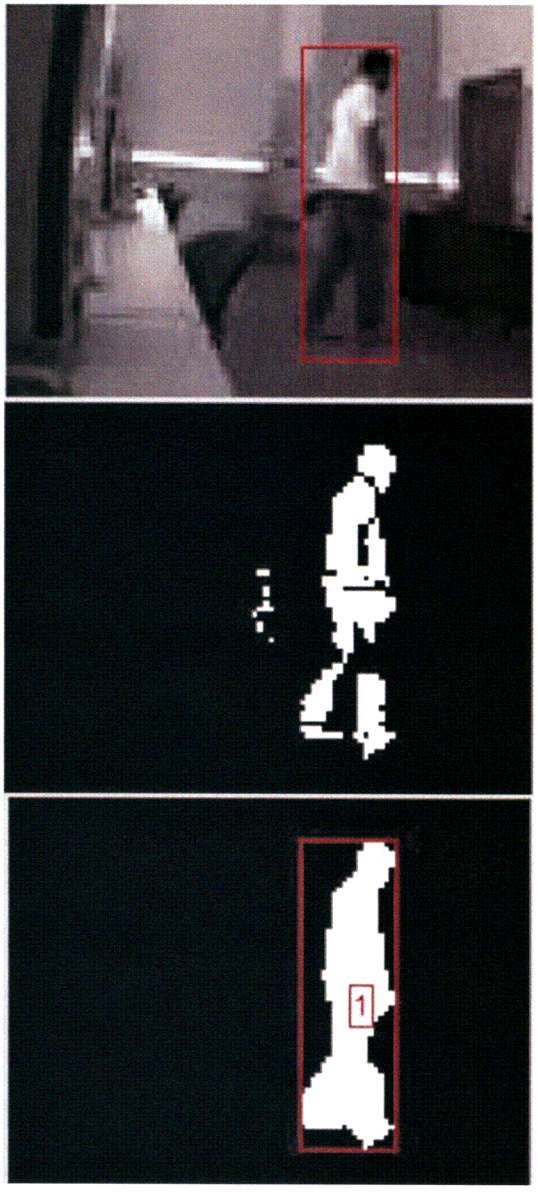

Figure 28. Bounded foreground object (top), foreground segmentation output (middle), and processed and enhanced foreground for "Cam3" sequence.

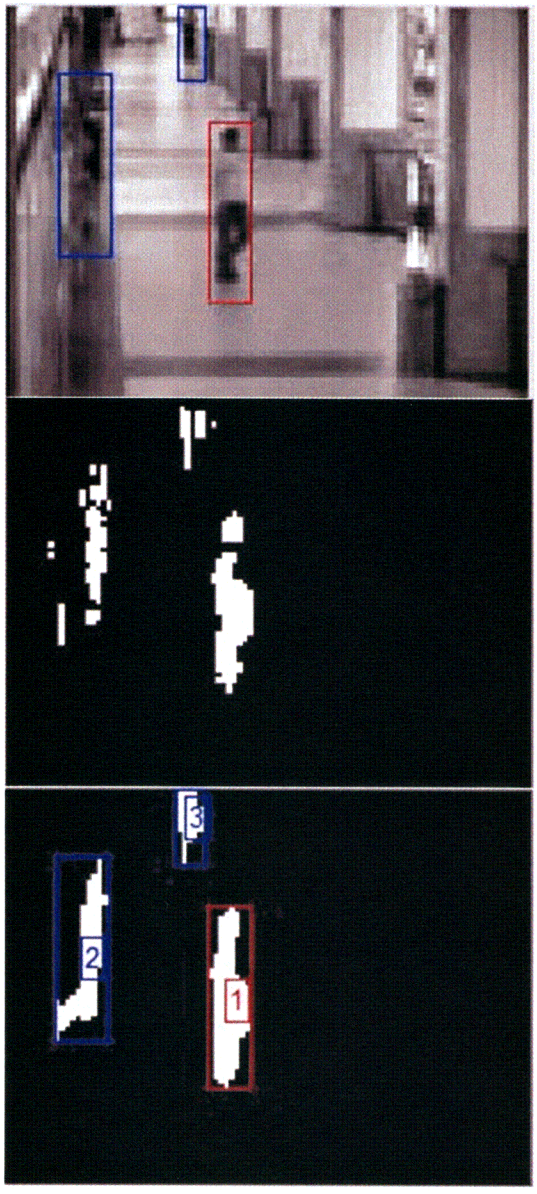

Figure 29. Bounded foreground object (top), foreground segmentation output (middle), and processed and enhanced foreground for "OneLeaveShopReenter1cor" sequence.

Figure 30. Bounded foreground object (top), foreground segmentation output (middle), and processed and enhanced foreground for "Black_Smoke_Masked" sequence.

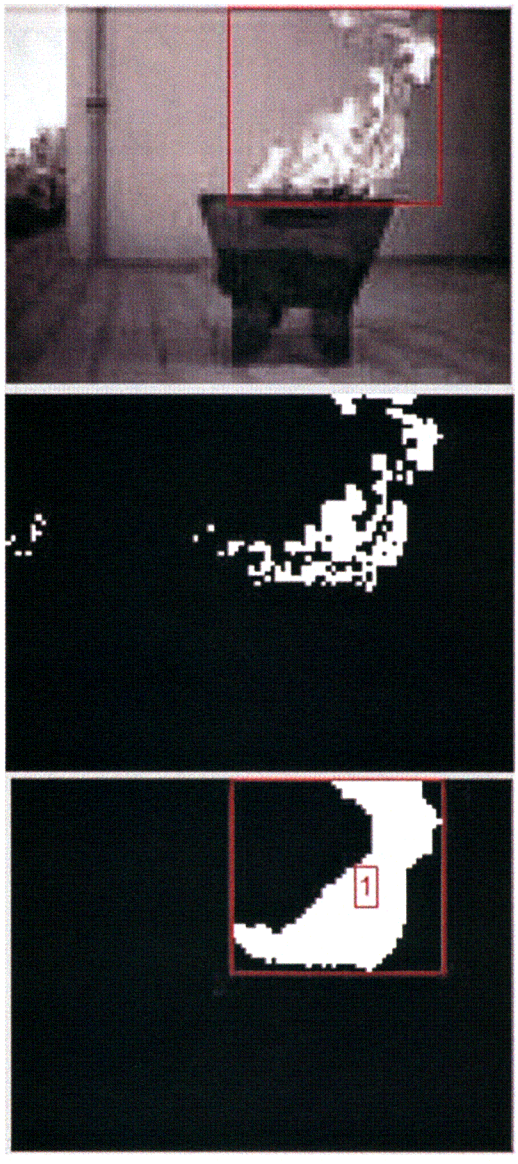

Figure 31. Bounded foreground object (top), foreground segmentation output (middle), and processed and enhanced foreground for "Barbeque" sequence.

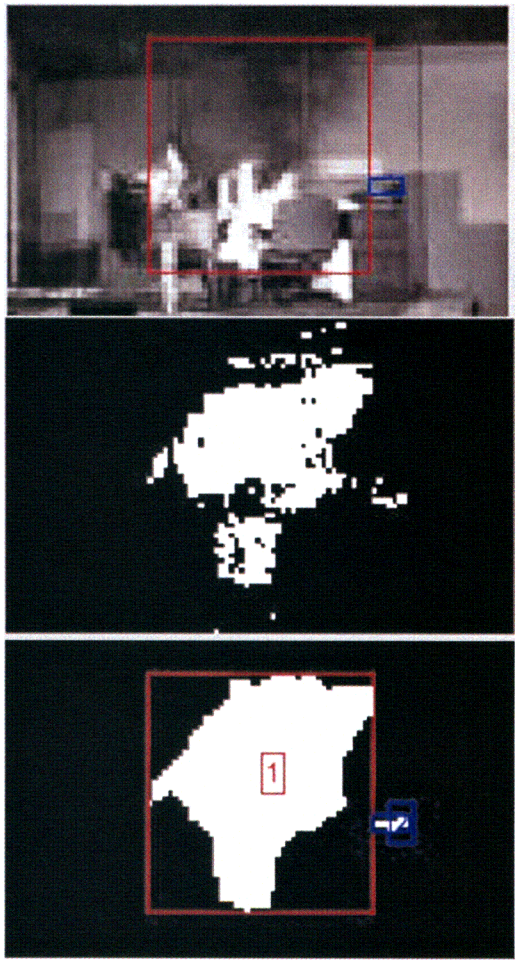

Figure 32. Bounded foreground object (top), foreground segmentation output (middle), and processed and enhanced foreground for "Single_Workstation" sequence.

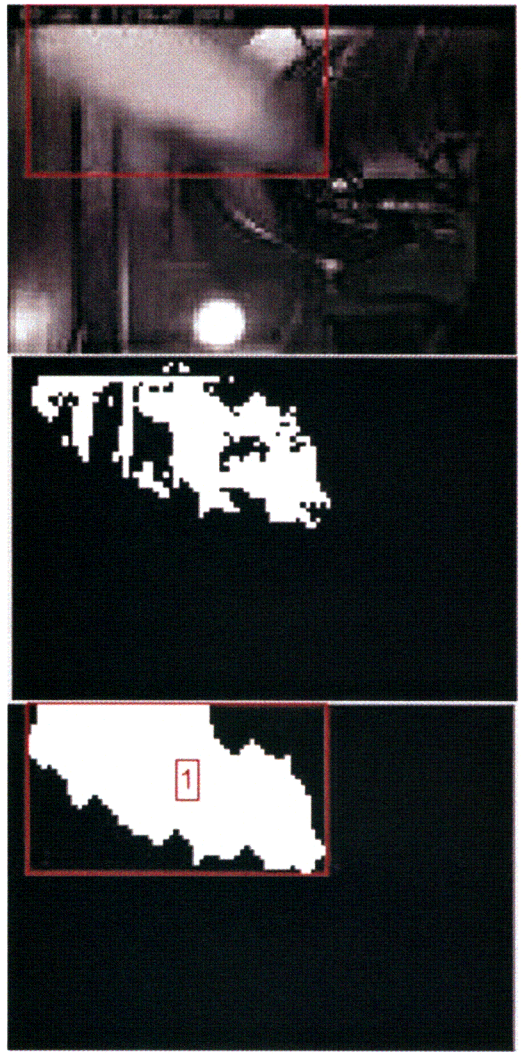

Figure 33. Bounded foreground object (top), foreground segmentation output (middle), and processed and enhanced foreground for "GTG12_particle_smoke" sequence.

4.5 Object Tracking

The ability to track foreground objects is critical to extracting time-dependent features for classification. The tracking system also provides for useful analysis of object positions in a sequence. The results shown in this section are primarily showing capabilities of the system and the effect of Kalman filtering foreground object centroid positions. In addition to plotting a bounding box around each foreground object, the feature extraction implementation also plots the position history of the foreground object that was tracked for the greatest number of successive frames. Figure 34 shows the tracked position history of a foreground object, which in this case is the flame of a burning outdoor barbeque. This clearly shows that the foreground object was tracked for quite some time. Immediately it is apparent that the object movement occurred in the region of object paths. The star symbol shows the birth of the foreground object, and the triangle shows the last point tracked in the sequence. The complete path of the object can be known at a single glance of the successful analysis result. The original object centroid position measurements can be seen by the dotted line, as well as the Kalman filtered centroid position. The Kalman filtered position provides a better estimate of the true position, minimizing measurement noise. A better example of the Kalman filtered trajectory can be seen in Figure 36, applied to a rising cloud of smoke.

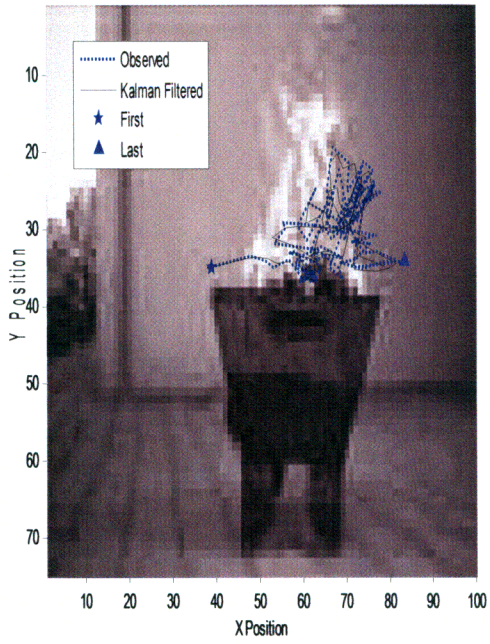

Figure 34. Tracked centroid position of foreground object in "Barbeque" sequence.

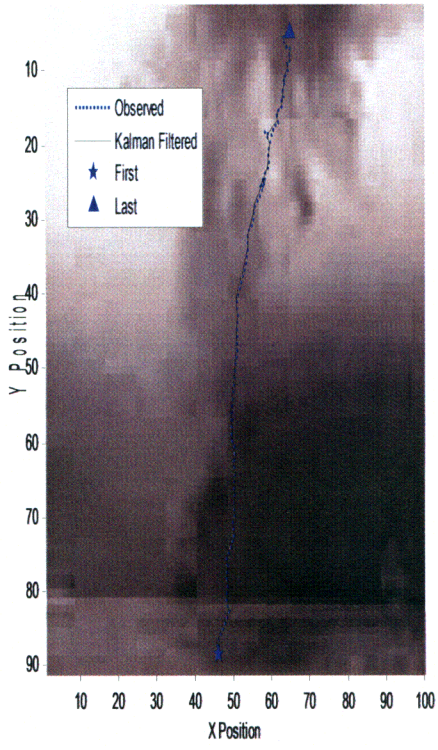

Figure 35. Tracked centroid position of foreground object in "Black_Smoke_Masked" sequence.

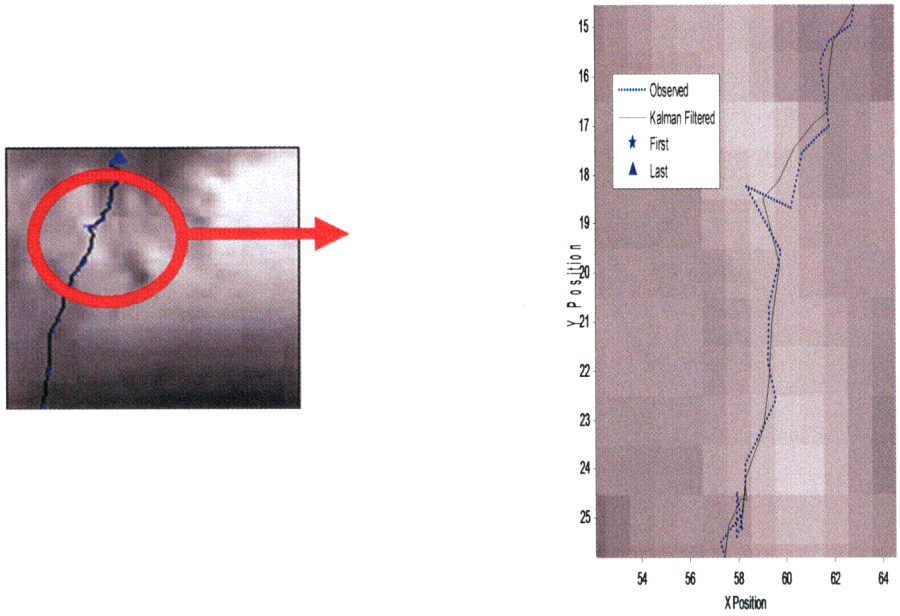

Figure 36. Zoomed foreground object position history for "Black_Smoke_Masked" sequence.

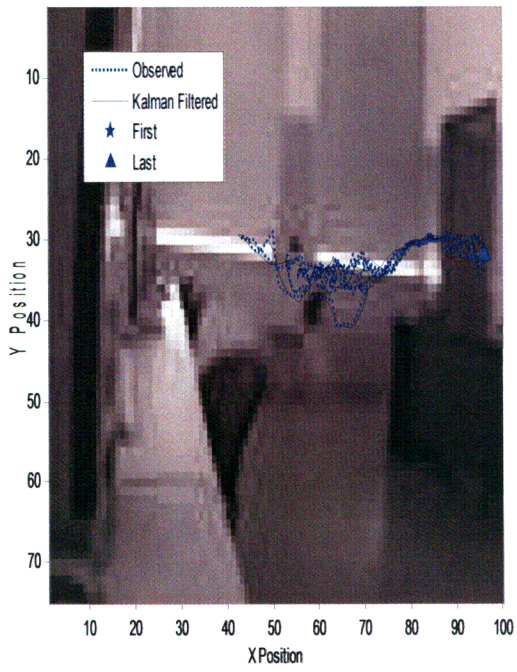

Figure 37. Tracked centroid position of foreground object in "Cam3" sequence.

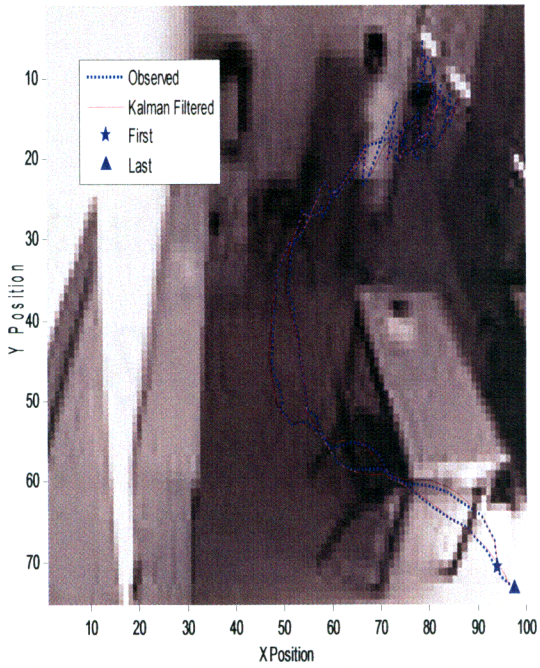

Figure 38. Tracked centroid position of foreground object in "Cam4" sequence.

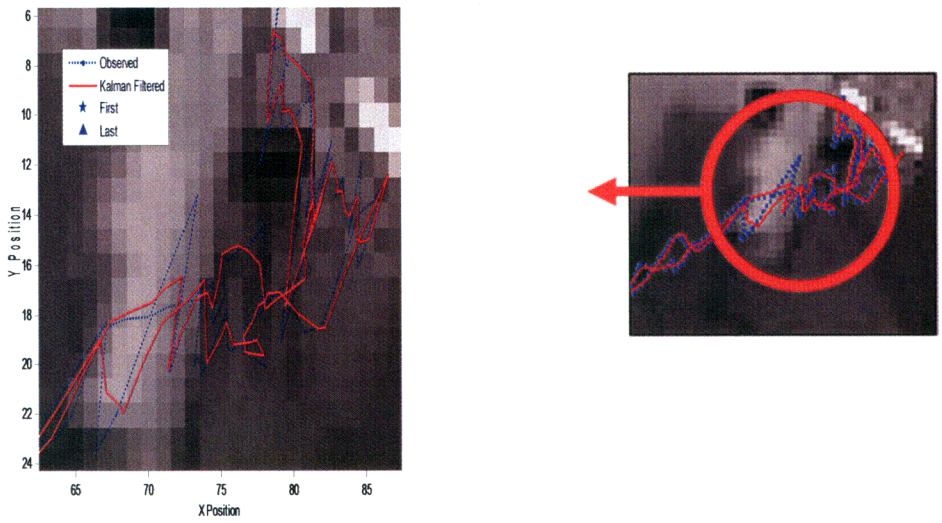

Figure 39. Zoomed foreground object position history in "Cam4" sequence.

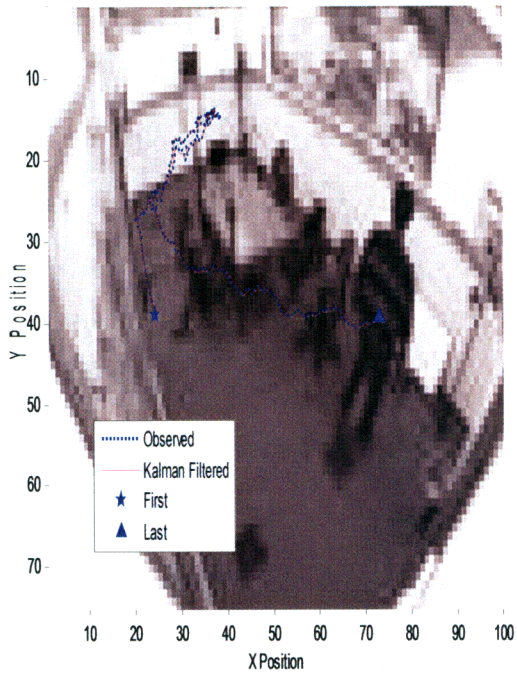

Figure 40. Tracked centroid position of foreground object in "intelligentroom_raw" sequence.

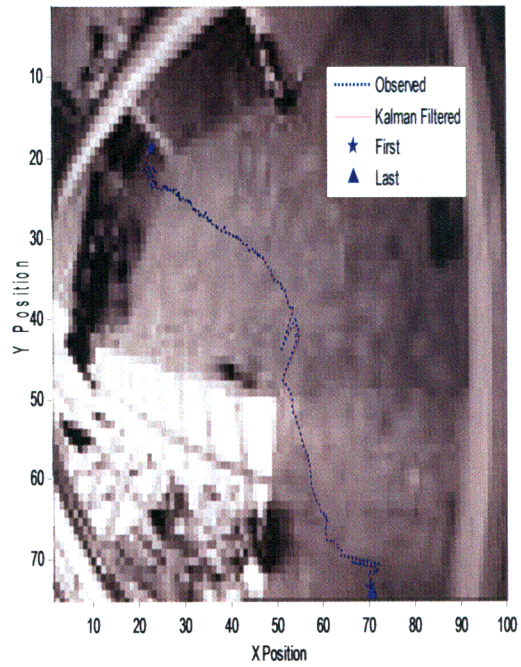

Figure 41. Tracked centroid position of foreground object in "Meet_Crowd" sequence.

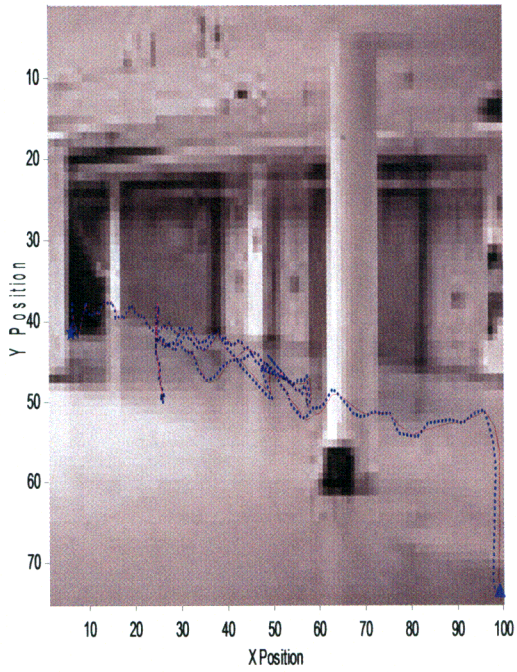

Figure 42. Tracked foreground object centroid position in "MSA" sequence.

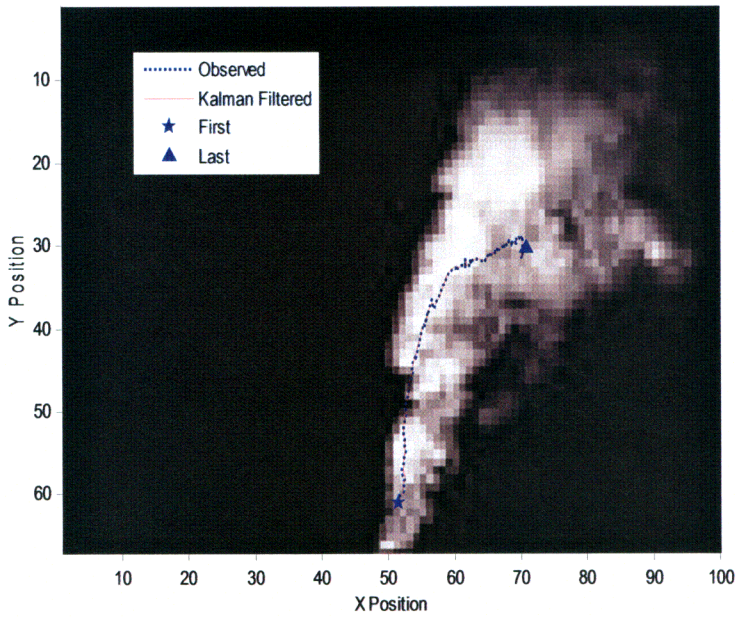

Figure 43. Tracked foreground object centroid position in "Smoke_Plume" sequence.

4.6 Object Feature Extraction

Features were extracted from the training database over all frames for the maximum specified number of objects tracked per frame (5 largest objects) that were tracked for at least 20 successive frames. The first analysis of the extracted training data features was a feature ranking analysis. The feature ranking was performed by training a neural network classifier using 10 hidden layer nodes and 200 epoch stopping criterion separately for each individual feature. The performances of training on the data set were ranked from best to worst. The results of this ranking can be seen in Table 14 and Table 15.

The results clearly show that the DCT features are most distinguishing. DCT features accounted for 19 of the top 30 ranked features. The top ranked feature was the first DCT coefficient of the masked gray level object. The first DCT coefficient is a measure of the average energy in the region. The conjecture that can be taken from this analysis is that the DCT coefficients have shown a great capability in capturing the low spatial frequency of smoke and fire anomalies in the training sequences. This is due to the fact that the first few DCT coefficients are effectively measures of low frequency content in an image.

After this analysis was performed, it was decided that the centroid position features (highlighted in red in Table 14) would not be used as part of the final classifier design. The centroid position showed reasonable performance compared to other calculated features, ranking at 13 and 15, however they are not good features to use for a general system. They are not invariant to translation in any manner, and are in fact entirely dependent on translation. This is undesirable because the training data being

used does not come from the exact scene context that the final system will be applied to. The classifier needs to be generalized so that no matter where activity occurs in the camera's FOV, it will properly distinguish between nuisance and anomalous events. If the final installation site contained the exact same scene context as the training data, then positional information would be a desirable feature. This would allow areas of high traffic for example to be favored heavily as being a nuisance objects when detected.

A noteworthy aspect is that the last few ranked features had performances very near or exactly 50%. This performance is no better than the result of a random guess between either an anomalous or nuisance classification. The Hu moment features did not perform well overall in particular. The growth feature (area change), which was believed to provide discriminating information in capturing the growth of smoke and fire regions, showed to be one of the worst performing features. The reason for this may have been the lack of a method of normalizing the change over time. This modification would effectively determine a rate of change rather than a raw change from the previous frame. The training database videos had inconsistent FPS parameters, meaning that the time interval between frames was not consistent for all videos. This was an oversight that should be corrected in future work. Also, based on this analysis it was determined that 40 out of the remaining 56 features would be retained after PCA was performed on the data set. This was decided based on the poor performance of the last 16 features.

Rank	Feature Name	Performance
1	Gray Level DCT 1	80.165%
2	Major Axis Length	76.896%
3	Perimeter	75.957%
4	Minor Axis Length	75.923%
5	Bounding Box Width	75.907%
6	Gray Level DCT 3	75.190%
7	Area	74.840%
8	Gray Level DCT 8	74.421%
9	Gray Level DCT 2	74.166%
10	Bounding Box Height	74.164%
11	Gray Level DCT 5	73.504%
12	Gray Level DCT 7	73.023%
13	Kalman Filtered Centroid X	72.338%
14	Binary DCT 3	71.740%
15	Kalman Filtered Centroid Y	71.578%
16	Gray Level DCT 6	71.252%
17	Binary DCT 7	70.885%
18	Gray Level DCT 10	70.818%
19	Compactness	70.554%
20	Binary DCT 2	70.335%
21	Binary DCT 8	70.299%
22	Gray Level DCT 9	70.178%
23	Binary DCT 10	69.847%
24	Binary DCT 5	69.624%
25	Binary DCT 6	69.392%
26	Gray Level DCT 4	69.384%
27	Binary DCT 9	67.388%
28	Gray Level Hu Moment 1	66.805%
29	Binary DCT 4	66.185%
30	Eccentricity	62.033%
31	Gray Level Stat. Skewness	62.019%
32	Binary Hu Moment 1	61.840%
33	Delta Y	61.501%
34	Average Gray Level	61.399%
35	Extent	60.866%
36	Orientation	60.051%

Table 14. Feature ranking results on training data set (1 to 36).

Rank	Feature Name	Performance
37	Y Velocity	60.023%
38	Gray Level Stat. Smoothness	59.577%
39	Gray Level Stat. Uniformity	59.548%
40	Gray Level Stat. Avg. Entropy	59.442%
41	Average Contrast	59.429%
42	Binary Hu Moment 2	59.058%
43	Delta X	55.216%
44	Binary Hu Moment 3	54.516%
45	X Velocity	52.091%
46	Gray Level Hu Moment 2	50.826%
47	Gray Level Hu Moment 4	50.670%
48	Gray Level Hu Moment 6	50.247%
49	Binary Hu Moment 4	50.081%
50	Gray Level Hu Moment 3	50.027%
51	Binary Hu Moment 5	50.020%
52	Binary Hu Moment 6	50.006%
53	Area Change	50.000%
54	Aspect Ratio	50.000%
55	Binary DCT 1	50.000%
56	Gray Level Hu Moment 5	50.000%
57	Gray Level Hu Moment 7	50.000%
58	Binary Hu Moment 7	50.000%

Table 15. Feature ranking results on training data set (37 to 58).

4.7 Object Classification

Perhaps the most important result is the performance of the final neural network classifier on the features extracted from the training database. K-fold cross validation was employed to get a generalized performance of the classifier on the training data. K-fold cross validation is performed by dividing the entire training data set into K randomly divided subsets of size N/K, where N is the total number of feature vectors in the training data set. The classifier is trained on all subsets except for one hold-out set. The performance is tested on the hold-out set and saved. This same process is repeated for a

total of K iterations, each time using a different hold-out set [97]. All of the K saved performances are then averaged, and a confidence interval is generated. The confidence interval is based on either a cumulative Student's t distribution critical t-score value with $K-1$ degrees of freedom (for $K<30$) or a standard normal/Gaussian cumulative distribution critical z-score value (for $K \geq 30$), both based on a desired level of confidence. The level of confidence, given by $100(1-\alpha)\%$, used in this thesis was 95% ($\alpha = 0.05$). The generated confidence intervals are given by

$$\text{Performance} = \begin{cases} \bar{\text{p}} \pm t_{\alpha/2,K-1}\dfrac{s}{\sqrt{K}}, & K<30 \\ \bar{\text{p}} \pm z_{\alpha/2}\dfrac{s}{\sqrt{K}}, & K \geq 30 \end{cases}, \tag{84}$$

where $\bar{\text{p}}$ is the average performance over all K validation tests and s is the sample standard deviation of the K performances [103]. As the value of K is increased, up to the total number of training data set samples, an increasingly better generalization of the true classifier performance is measured.

The training data set referred to in this section is the entire set of extracted features based on exercising feature extraction on the entire training video database. The set of feature vectors used for training do not carry any specific correlation to which frame or video they originated. Each feature vector was calculated for one foreground object in an unspecified frame (of which there may be multiple foreground objects contained) of one of the unspecified videos of the training video database. Feature vectors are tied to the video database only by their class label which may be either anomalous or nuisance. This means that performance is not given on a per-frame basis,

but instead given for those objects detected as part of the foreground for frames actually containing detected foreground movement. This was considered to be a fair judgment of performance, given that foreground objects will nearly always be detected given correct calibration of the system for a video source. In other words, false positives are of much greater occurrence rather than false negatives since the system would not have any event to actually classify in a frame containing no movement or insignificant movement.

Performance of the final neural network classifier is displayed in Table 16. The two layer neural network architecture used in this thesis contains one hidden layer with 5 hidden layer nodes. The activation transfer function used was the logistic sigmoid for the hidden layer nodes and the output layer nodes.

Conditions	Average Performance ($\bar{p}$)	Confidence Interval	Standard Deviation (s)
5-fold cross validation	95.75%	{95.42%, 96.07%}	1.95%
10-fold cross validation	95.90%	{95.67%, 96.13%}	1.85%
25-fold cross validation	95.58%	{95.41%, 95.74%}	2.08%
50-fold cross validation	94.83%	{94.01%, 95.65%}	2.97%
100-fold cross validation	95.43%	{95.13%, 95.72%}	1.51%

Table 16. Neural network average classifier performance using K-fold cross validation with 95% confidence.

CHAPTER 5: CONCLUSIONS

This thesis has focused on the design, development and validation of algorithms for the detection and tracking of anomalous events that can be identified from the analysis of monochromatic stationary ship surveillance video streams. The specific anomalies that have been focused upon are the presence and growth of smoke and fire events inside the frames of video streams.

5.1 Summary of Accomplishments

The objectives of this thesis are revisited below with an explanation of how they were accomplished:

1. Compilation of a survey of existing techniques for analyzing shipboard video stream data;

   A survey of existing video analysis techniques has been compiled as previously discussed in section 2.1. This literature survey encompassed not only works in the area of video smoke and fire detection, but also that of general surveillance video analysis systems.

2. Design and development of a video foreground segmentation algorithm for determining regions of interest in video streams;

   The literature survey provided a springboard for the development of the foreground segmentation algorithm used in this thesis. Based on analyzing the available methods of foreground segmentation, it was determined for this application that

statistical background modeling was the best candidate. The adaptive GMM based algorithm was implemented as the first method of extracting the moving regions of source video streams.

3. Design and development of an object tracking system capable of persistently tracking objects between frames;

The literature survey also provided basis for the development of the object tracking system developed for this thesis. Foreground segmented objects are successfully tracked between frames, with their position measurements applied to a Kalman filter to provide an estimate of the true position with reduced measurement uncertainty and noise. The tracking system was employed to allow for the calculation of spatiotemporal features exhibited by foreground objects.

4. Identification of distinct and robust features from the tracked objects for detection and classification of anomalous indications;

A variety of features of the tracked foreground objects have been calculated, including shape features, spatiotemporal features, statistical features, and spectral features. The independent performance for each feature in distinguishing between nuisance and anomalous events has been calculated, identifying well-suited features for the task of smoke and fire event classification in light of nuisance events.

5. Execution of the algorithm on a database consisting of canonical and experimental videos streams embedded with known anomalies (smoke and fire) as well as benign content;

This entire surveillance video analysis system has been applied to a compiled database of videos for the training of a neural network classifier capable of distinguishing

between anomalous and nuisance events in monochromatic surveillance video streams. This system has been shown through cross validation classifier testing to perform well on classification of the diverse set of data presented to the algorithm.

As with any work, this thesis approach does have some drawbacks and could be improved in a few ways. The approach relies heavily on the availability of a large and diverse set of training videos. Relying on training data to provide an estimate of the true distribution of features discriminating between anomalous and nuisance events can be a challenge, especially in a life-critical situation such as smoke and fire detection. Training data must be chosen very carefully in order to be confident that the classifier is being trained on features representative of what is desired to be detected. Fortunately, increasing the size of the training database should theoretically decrease this risk by giving the classifier a more diverse and larger set of feature samples to examine during the training process.

Due to the many techniques used in the approach of this work, there also exists a large number of parameters required for operation. Many of these parameters will not change depending on the video stream being analyzed, but some of those related to the foreground segmentation and enhancement may require adjustment depending on image quality and resolution. Care was taken in normalizing the features calculated for foreground objects, but optimizing parameters requiring adjustment may be take some time and will depend on final system installation or training video selection.

The foreground segmentation algorithm method used, although adjustable for varying scenarios, is computationally intensive (particularly for large images) and shows

promise for a dedicated hardware implementation. The training videos used in this work had all been resized to 100 pixels width prior to analysis in order to expedite processing. This shows that this smaller size image, which is effectively a lower resolution image, still provided distinguishing features to classify tracked anomalies in video.

5.2 Recommendations for Future Work

Future work on this thesis may explore the use of other image features such as wavelet coefficients, windowed averages of object growth, and windowed averages of velocity measurements. Also, although extracted features were normalized as much as possible in terms of their calculation, it may be beneficial to normalize all foreground objects by resizing them to specified height and width prior to feature extraction, guaranteeing that each object is treated equally. The foreground segmentation algorithm could possibly be optimized for parallel processing to decrease processing time. Tracking could also be improved by adding occlusion detection with robust merge/split handling for foreground objects. Classifier parameters could certainly be optimized further, as classification performance was not a primary objective of this thesis. It may be beneficial to explore the use of types of classifiers other than neural networks. Also, since videos were resized to 100px width prior to analysis, possible improvement in classification performance could be determined for resizing images to a greater size, capturing more information in each frame. Finally, as mentioned previously, the training video database could be expanded to provide a greater diversity of extracted features for both anomalous events as well as nuisance events.

This thesis has been comprised of research and techniques enabling the detection of smoke and fire anomalous events in shipboard video. Foreground segmentation and tracking have been applied to facilitate the task of feature extraction. A variety of features have been calculated for segmented motion, allowing the algorithm to determine exactly the features representative of anomalous conditions in experimental video streams in the compiled database. The trained neural network based on features extracted using these algorithms has shown to be successful in its ability to discriminate between anomalous events and nuisance events in shipboard video streams. It is the author's hope that this work can be continued in the future to accommodate additional anomalous events as well as additional features describing them.

REFERENCES

[1] J. B. Hinkle and T. L. Glover, "Reduced manning in DDG51 class warships: challenges, opportunities and the way ahead for reduced manning on all United States Navy ships," *Executive Assessment Prepared for PEO Ships*, Anteon Corp., Arlington, Virginia, 2004.

[2] J. B. Hinkle and T. L. Glover, "DDG-51 reduced manning study – phase I the concept: executive assessment and DDG-51 reduced manning study," *Executive Assessment Prepared for PEO Ships*, Anteon Corp., Arlington, Virginia, 2003.

[3] T. Schultze, T. Kempka, and I. Willms, "Audio–video fire-detection of open fires," *Fire Safety Journal*, vol. 41, no. 4, pp. 311–314, 2006.

[4] J. A. Neal, C. E. Land, R. R. Avent, and R. J. Churchill, "Application of artificial neural networks to machine vision flame detection," *Final Report*, American Research Corporation of Virginia, Radford, Virginia, 1991.

[5] B. U. Töreyin and others, "Computer vision based method for real-time fire and flame detection," *Pattern Recognition Letters*, vol. 27, no. 1, pp. 49–58, 2006.

[6] B. C. Ko, K. H. Cheong, and J. Y. Nam, "Fire detection based on vision sensor and support vector machines," *Fire Safety Journal*, vol. 44, no. 3, pp. 322–329, 2009.

[7] D. Han and B. Lee, "Flame and smoke detection method for early real-time detection of a tunnel fire," *Fire Safety Journal*, vol. 44, no. 7, pp. 951–961, 2009.

[8] T. Ono et al., "Application of neural network to analyses of CCD colour TV-camera image for the detection of car fires in expressway tunnels," *Fire Safety Journal*, vol. 41, no. 4, pp. 279–284, 2006.

[9] C. Ho, "Machine vision-based real-time early flame and smoke detection," *Measurement Science and Technology*, vol. 20, no. 4, p. 045502, 2009.

[10] D. Kim and Y. Wang, "Smoke detection in video," *Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering*, pp. 759-763, Los Angeles, California, 2009.

[11] S. Calderara, P. Piccinini, and R. Cucchiara, "Smoke detection in video surveillance: a MoG model in the wavelet domain," *Computer Vision Systems*, pp. 119-128, vol. 5008, Springer Berlin/Heidelberg, Berlin/Heidelberg, Germany, 2008.

[12] J. Gubbi, S. Marusic, and M. Palaniswami, "Smoke detection in video using wavelets and support vector machines," *Fire Safety Journal*, vol. 44, no. 8, pp. 1110–1115, 2009.

[13] F. Gomez-Rodriguez, B. Arrue, and A. Ollero, "Smoke monitoring and measurement using image processing: Application to forest fires," *Proceedings of the Eighth SPIE Conference on Automatic Target Recognition XIII*, pp. 404-411, Orlando, Florida, 2003.

[14] Y. Chunyu, F. Jun, W. Jinjun, and Z. Yongming, "Video fire smoke detection using motion and color features," *Fire Technology*, vol. 46, no. 3, pp. 651-663, 2009.

[15] C. Liu and N. Ahuja, "Vision based fire detection," *Proceedings of the 17th International Conference on Pattern Recognition*, pp. 134-137, Cambridge, UK, 2004.

[16] R. Collins et al., "A system for video surveillance and monitoring," *Proceedings of the American Nuclear Society (ANS) Eighth International Topical Meeting on Robotics and Remote Systems*, pp. 25–29, Pittsburgh, PA, 1999.

[17] M. Cristani, M. Bicego, and V. Murino, "Audio-visual event recognition in surveillance video sequences," *IEEE Transactions on Multimedia*, vol. 9, no. 2, pp. 257-267, 2007.

[18] M. Brand and V. Kettnaker, "Discovery and segmentation of activities in video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 844–851, 2000.

[19] G. Medioni, I. Cohen, F. Brémond, S. Hongeng, and R. Nevatia, "Event detection and analysis from video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 873–889, 2001.

[20] T. Xiang and S. Gong, "Video behavior profiling for anomaly detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 893-908, 2008.

[21] D. Koller et al., "Towards robust automatic traffic scene analysis in real-time," *Proceedings of the 12th International Conference on Pattern Recognition*, pp. 126-131, Jerusalem, Israel, 1994.

[22] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, 2000.

[23] T. Xiang, S. Gong, and D. Parkinson, "Autonomous visual events detection and classification without explicit object-centred segmentation and tracking," *Proceedings of the British Machine Vision Conference*, pp. 233–242, Cardiff, UK, 2002.

[24] Z. Zhang, M. Li, K. Huang, and T. Tan, "Boosting local feature descriptors for automatic objects classification in traffic scene surveillance," *Proceedings of the 19th International Conference on Pattern Recognition*, pp. 1–4, Tampa, Florida, 2008.

[25] J. M. Gryn, R. P. Wildes, and J. K. Tsotsos, "Detecting motion patterns via direction maps with application to surveillance," *Computer Vision and Image Understanding*, vol. 113, no. 2, pp. 291–307, 2009.

[26] M. Jager, C. Knoll, and F. Hamprecht, "Weakly supervised learning of a classifier for unusual event detection," *IEEE Transactions on Image Processing*, vol. 17, no. 9, pp. 1700-1708, 2008.

[27] C. Stauffer and W. E. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[28] A. Basharat, A. Gritai, and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, Anchorage, Alaska, 2008.

[29] J. Muuss, *The Complete Guide for CPP Examination Preparation,* pp. 223-232, Auerbach Publications, Boca Raton, 2006.

[30] W. Grondzik, *Mechanical and Electrical Equipment for Buildings*, 11th ed., pp. 1143-1153, Wiley, Hoboken, New Jersey, 2010.

[31] R. Zalosh and P. Chantranuwat, "International road tunnel fire detection research project phase I - review of prior test programs and tunnel fires," *Technical Report*, The Fire Protection Research Foundation, Quincy, Massachusetts, 2003.

[32] D. T. Gottuk et al., "An initial evaluation of video-based fire detection technologies," *Memorandum Report*, Naval Research Laboratory, Washington, DC, 2004.

[33] *NFPA 72: National Fire Alarm and Signaling Code.* <http://www.nfpa.org/aboutthecodes/AboutTheCodes.asp?DocNum=72&cookie%5Ftest=1>, accessed July 15, 2010.

[34] D. T. Gottuk, "Video image detection systems installation performance criteria," *Technical Report*, Hughes Associates, Inc., Baltimore, Maryland, 2008.

[35] *Underwriters Laboratories Standard 268B Scope*, <http://ulstandardsinfonet.ul.com/outscope/0268B.html>, accessed July 16, 2010.

[36] *Underwriters Laboratories Standard 268B Outline,* <http://ulstandardsinfonet.ul.com/tocs/tocs.asp?doc=o&fn=o0268B.toc>, accessed July 16, 2010.

[37] G. Privalov and J. A. Lynch, "Video image detection systems for fire and smoke (implementation and testing)," *Technical Report*, AxonX, Sparks, Maryland, 2007.

[38] S. L. Rose-Pehrsson et al., "Early warning fire detection system using a probabilistic neural network," *Fire Technology*, vol. 39, no. 2, pp. 147–171, 2003.

[39] S. L. Rose-Pehrsson et al., "Volume sensor for damage assessment and situational awareness," *Fire Safety Journal*, vol. 41, no. 4, pp. 301–310, 2006.

[40] S. L. Rose-Pehrsson, J. C. Owrutsky, D. T. Gottuk, J. A. Geiman, F. W. Williams, and J. P. Farley, "Phase I - FY01 investigative study for the advanced volume sensor," *Interim Report*, Naval Research Laboratory, Washington, DC, 2003.

[41] D. T. Gottuk, J. A. Lynch, S. L. Rose-Pehrsson, J. C. Owrutsky, and F. W. Williams, "Video image fire detection for shipboard use," *Fire Safety Journal*, vol. 41, no. 4, pp. 321–326, 2006.

[42] B. Ellaschuk and B. Lienert, "Critical assessment of damage/fire control systems and technologies for naval vessels in support of damage control and crew optimization: risks and opportunities phase IIa: fire suppression systems and components," *Contract Report*, Defence Research and Development Canada, Ottawa, Ontario, Canada, 2007.

[43] R. Beltowski, "Fire aboard a ship of war," *Fire Engineering*, vol. 150, no. 11, p. 73, Nov-1997.

[44] Committee on Assessment of Fire Suppression Substitutes and Alternatives to Halon, Commission on Physical Sciences, Mathematics, and Applications, National Research Council, *Fire suppression substitutes and alternatives to halon for U.S. Navy applications*, pp. 3-19, 45-48, National Academy Press, Washington DC, 1997.

[45] *NAVSEA Damage Control, Fire Protection Engineering and CBR-D Carbon Dioxide System*, <http://www.dcfpnavymil.org/dcc/equip/co2sys.htm>, accessed August 5, 2010.

[46] R. G. Bill Jr., R. L. Hansen, and K. Richards, "Fine-Spray (water mist) protection of shipboard engine rooms," *Fire Safety Journal*, vol. 29, no. 4, pp. 317–336, 1997.

[47] *Navy Technology Center for Safety & Survivability Damage Control Systems - Shipboard Fire Scaling*, <http://www.nrl.navy.mil/chemistry/6180/6186/damage control.php>, accessed August 5, 2010.

[48] W. Gatchell, "Shipboard firefighting: the basics," *Fire Engineering*, pp. 133-136, August 2003.

[49] R. A. Colombi Jr., "Ship fires vs. structure fires: differences and preparation," *Fire Engineering*, pp. 79-84, November 2009.

[50] L. C. Baucom, "Navy occupational safety and health (NAVOSH) program manual for forces afloat: volume I," Program manual, Department of the Navy, Washington, DC, 2000.

[51] *Navy Technology Center for Safety & Survivability Shipboard Fire Scaling*, <http:// www.nrl.navy.mil/chemistry/6180/6186/index.php>, accessed August 5, 2010.

[52] *Navy Technology Center for Safety & Survivability Chesapeake Bay Fire Test Detachment*, <http://www.nrl.navy.mil/chemistry/6180/6180.1/index.php>, accessed August 5, 2010.

[53] M. Piccardi, "Background subtraction techniques: a review," *Presented at the ARC Centre of Excellence for Autonomous Systems*, 2004. <http://www.staff.it.uts.edu. au/~massimo/BackgroundSubtractionReview-Piccardi.pdf>, accessed July 28, 2010.

[54] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1, pp. 185–203, 1981.

[55] D. H. Warren and E. R. Strelow, *Electronic Spatial Sensing for the Blind: Contributions from Perception, Rehabilitation, and Computer Vision*, NATO Scientific Affairs Division, Boston, 1985.

[56] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43-77, 1994.

[57] J. M. Bodily, "An optical flow implementation comparison study," *Masters Thesis*, Brigham Young University, Provo, Utah, 2009.

[58] *Wolfram MathWorld - Taylor Series*, <http://mathworld.wolfram.com/TaylorSeries. html> accessed June 13, 2010.

[59] J. Barron, "The 2D/3D differential optic flow," *Presented at the Canadian Conference on Computer and Robot Vision*, Kelowna, British Columbia, May 2009, <http://computerrobotvision.org/2009/tutorial_day/CRV2009diffopticalflow.pdf>, accessed June 14, 2010.

[60] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proceedings of the International Joint Conference on Artificial Intelligence*, vol. 3, p. 3, Vancouver, Canada, 1981.

[61] J. L. Barron and N. A. Thacker, "Tutorial: computing 2D and 3D optical flow," Memorandum report, University of Manchester, Manchester, UK, 2005.

[62] D. J. Fleet and A. D. Jepson, "Computation of component image velocity from local phase information," *International Journal of Computer Vision*, vol. 5, no. 1, pp. 77-104, 1990.

[63] A. M. Waxman, J. Wu, and F. Bergholm, "Convected activation profiles and the measurement of visual motion," *Proceedings of The Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 717-723, Ann Arbor, Michigan, 1988.

[64] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," *International Journal of Computer Vision*, vol. 2, no. 3, pp. 283-310, 1989.

[65] A. Singh and P. Allen, "Image-flow computation: an estimation-theoretic framework and a unified perspective," *CVGIP: Image Understanding*, vol. 56, no. 2, pp. 152-177, 1992.

[66] D. J. Heeger, "Model for the extraction of image flow," *Journal of the Optical Society of America A*, vol. 4, no. 8, p. 1455, 1987.

[67] H. Liu, T. Hong, M. Herman, and R. Chellappa, "A general motion model and spatio-temporal filters for computing optical flow," *International Journal of Computer Vision*, vol. 22, no. 2, pp. 141-172, 1997.

[68] T. Camus, "Real-time quantized optical flow," *Journal of Real Time Imaging*, vol. 3, pp. 71–86, 1997.

[69] E. P. Simoncelli, "Bayesian multi-scale differential optical flow," *Handbook of Computer Vision and Applications*, Academic Press, 1999, pp. 397-422.

[70] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[71] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 142–149, Hilton Head Island, South Carolina, 2000.

[72] R. M. Haralick, "Propagating covariance in computer vision," *Performance Characterization in Computer Vision*, p. 95, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[73] R. Marik, J. Kittler, and M. Petrou, "Error sensitivity assessment of vision algorithms based on Direct Error-Propagation," *Proceedings of the ECCV Workshop on Performance Characteristics of Vision Algorithms*, Cambridge, UK, 1996.

[74] A. Bovik, *The Essential Guide to Video Processing*, 2nd ed., pp. 442-449, Academic Press/Elsevier, Boston, 2009.

[75] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780-785, 1997.

[76] C. Bishop, *Pattern Recognition and Machine Learning*, 1st ed., pp. 435-439, Springer, New York, 2006.

[77] C. Stauffer and W. E. Grimson, "Adaptive background mixture models for real-time tracking," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 246–252, Fort Collins, Colorado, 1999.

[78] P. W. Power and J. A. Schoonees, "Understanding background mixture models for foreground segmentation," *Proceeedings of the Image and Vision Computing New Zealand*, pp. 267-271, Auckland, New Zealand, 2002.

[79] C. Ridder, O. Munkelt, and H. Kirchner, "Adaptive background estimation and foreground detection using Kalman filtering," *Proceedings of the International Conference on Recent Advances in Mechatronics*, pp. 193–199, Istanbul, Turkey, 1995.

[80] K. Karmann and A. V. Brandt, "Moving object recognition using an adaptive background memory," *Time-varying Image Processing and Moving Object Recognition*, vol. 2, pp. 297-307, 1990.

[81] M. Boninsegna and A. Bozzoli, "A tunable algorithm to update a reference image," *Signal Processing: Image Communication*, vol. 16, no. 4, pp. 353-365, 2000.

[82] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," *Proceedings of the International Conference on Computer Vision*, vol. 1, p. 29, Kerkyra, Greece, 1999.

[83] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1163, 2002.

[84] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1168-1177, 2008.

[85] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground–background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172-185, Jun. 2005.

[86] D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht, "Neural network approach to background modeling for video object segmentation," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1614-1627, 2007.

[87] S. C. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," *Video Communications and Image Processing, SPIE Electronic Imaging*, vol. 5308, pp. 881–892, 2004.

[88] R. Gonzalez and R. Woods, *Digital Image Processing*, 2nd ed., pp. 472-473, 520-531, 672-675, 666-668, Prentice Hall, Upper Saddle River, New Jersey, 2002.

[89] M. S. Grewal and A. P. Andrews, *Kalman filtering: Theory and Practice Using MATLAB*, 2nd ed., pp. 1-5, Wiley-Interscience, New York, New York, 2001.

[90] G. Welch and G. Bishop, "An introduction to the Kalman filter," *Presented at SIGGRAPH*, Los Angeles, California, 2001 <http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf>, accessed July 29, 2010.

[91] M. Nixon and A. Aguado, *Feature Extraction and Image Processing*, pp. 278-285, Newnes, Oxford, UK, 2002.

[92] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. 1, pp. 73-76, 656-657, Addison-Wesley, Reading, Massachusetts, 1992.

[93] I. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*, pp. 46-50, Hoboken, Wiley, New Jersey, 2003.

[94] A. Bovik, *The Essential Guide to Image Processing*, 2nd ed., pp. 432-433, Academic Press, Burlington, Massachusetts, 2009.

[95] W. Pratt, *Digital Image Processing: PIKS Scientific Inside*, 4th ed., pp. 189-203, Wiley Interscience, Hoboken, New Jersey, 2007.

[96] R. Polikar, "Dimensionality reduction," *Advanced Topics in Pattern Recognition Class Lecture*, Department of Electrical and Computer Engineering, Rowan University, Glassboro, New Jersey, 2009.

[97] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed., pp. 483-485, 568, Wiley, New York, New York, 2001.

[98] R. Polikar, "The multilayer perceptron," *Advanced Topics in Pattern Recognition Class Lecture*, Department of Electrical and Computer Engineering, Rowan University, Glassboro, New Jersey, 2009.

[99] K. Murphy, *Kalman Filter Toolbox for Matlab*, <http://www.cs.ubc.ca/~murphyk/Software/Kalman/kalman.html>, 1998.

[100] R. P. W. Duin et al., *PRTools4.1, A Matlab Toolbox for Pattern Recognition*, <http://www.prtools.org/>, Delft University of Technology, 2007.

[101] R. Fisher, J. Santos-Victor, and J. Crowley, *CAVIAR: Context Aware Vision Using Image-Based Active Recognition*, <http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/>, accessed July 30, 2010.

[102] R. Vezzani and R. Cucchiara, "Video surveillance online repository (ViSOR): an integrated framework," *Multimedia Tools and Applications*, vol. 50, no. 2, pp. 359-380, 2009.

[103] R. Polikar, "Background," *Advanced Topics in Pattern Recognition Class Lecture*, Department of Electrical and Computer Engineering, Rowan University, Glassboro, New Jersey, 2009.